

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу  
Кафедра математичних методів системного аналізу**

До захисту допущено:

В.о.завідувача кафедри

\_\_\_\_\_ Оксана ТИМОЩУК

«\_\_» \_\_\_\_\_ 20\_\_ р.

**Дипломна робота**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Системний аналіз і управління»  
спеціальності 124 «Системний аналіз»**

**на тему: «Моделі і методи нелінійних нестационарних процесів в  
економіці та фінансах»**

Виконав (-ла):

студент (-ка) IV курсу, групи КА-63

Кукуруза Єлизавета Сергіївна

\_\_\_\_\_

Керівник:

Д.т.н, професор кафедри ММСА

Бідюк Петро Іванович

\_\_\_\_\_

Консультант з нормконтролю :

Доцент, к.т.н.

Коваленко А.Є.

\_\_\_\_\_

Рецензент:

Професор, д.т.т

Архипов Олександр Євгенович

\_\_\_\_\_

Засвідчую, що у цій дипломній роботі  
немає запозичень з праць інших авторів  
без відповідних посилань.

Студент (-ка) \_\_\_\_\_

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Інститут прикладного системного аналізу**

**Кафедра математичних методів системного аналізу**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 124 «Системний аналіз»

Освітньо-професійна програма «Системний аналіз і управління»

ЗАТВЕРДЖУЮ

В.о.завідувача кафедри

\_\_\_\_\_ Оксана ТИМОЩУК

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**на дипломну роботу студенту**

**Кукуруза Єлизавета Сергіївна**

1. Тема роботи «Моделі і методи нелінійних нестационарних процесів в економіці та фінансах», керівник роботи Бідюк Петро Іванович, д.т.н. , професор кафедри ММСА, затверджені наказом по університету від « 25 » травня 20 20 р. № 1143-с

2. Термін подання студентом роботи 08 травня 2020 року

3. Вихідні дані до роботи

4. Зміст роботи

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

6. Консультанти розділів роботи\*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Шевчук О.А., к.е.н., доцент		

7. Дата видачі завдання №1143-с від 25.05.2020.

#### Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Формулювання тематики (напрямку) дослідження.	03.09.2019 – 30.09.2019	виконано
2	Аналіз актуальності задач стосовно тематики дослідження	01.10.2019 – 30.10.2019	виконано

---

3	Аналіз відомих результатів стосовно тематики дослідження	01.11.2019 – 30.11.2019	виконано
4	Формулювання задач дослідження	01.12.2019 – 30.12.2019	виконано
5	Уточнення теми дипломної роботи	25.02.2019	виконано
6	Збір статичних даних, попередній аналіз даних	01.03.2020 – 30.03.2020	виконано
7	Розробка програмного продукту для виконання обчислювальних експериментів	01.03.2020 – 30.04.2020	виконано
8	Виконання обчислювальних експериментів, аналіз та оформлення результатів	01.05.2020 – 20.05.2020	виконано
9	Оформлення пояснювальної записки у цілому	21.05.2020 – 31.05.2020	виконано
10	Підготовка презентації для захисту	28.05.2020 – 01.06.2020	виконано
11	Попередній захист дипломної роботи	01.06.2020 – 03.06.2020	виконано
12	Захист дипломної роботи	15.06.2020 – 18.06.2020	виконано

Студент

Єлизавета Кукуруза

Керівник

Петро Бідюк

## РЕФЕРАТ

Дипломна робота: 111 с., 18 рис., 18 табл., 2 дод., і 20 джерел.

МАТЕМАТИЧНА МОДЕЛЬ, КРИТЕРІЇ ЯКОСТІ, КРИТЕРІЇ  
АДЕКВАТНОСТІ, ГЕТЕРОСКЕДАСТИЧНІСТЬ, ІНТЕГРОВАНІСТЬ.

Об'єкт дослідження: нелінійні нестационарні процеси в економіці та фінансах.

Мета роботи: порівняти методи і моделі прогнозування на основі економічних даних.

Предмет дослідження: Методи і моделі опису процесів в економіці.

В роботі досліджуються поведінка економічних показників, їх аналіз та прогнозування. Процеси фінансового ринку, які по своїй природі є нелінійними та нестационарними, сьогодні піддаються бурхливому аналізу. Тому для дослідження таких процесів потрібно використовувати відповідні моделі та методи для оцінювання їх параметрів. В роботі досліджено ряд моделей, що описують нелінійні нестационарні процеси.

## ABSTRACT

Thesis: 111 p., 30 fig., 18 tabl., 2 append., 20 sources.

MATHEMATICAL MODEL, CRITERIA OF QUALITY, ADEQUACY  
CRITERIA, HETEROSKEDASTICITY, INTEGRATION.

The paper investigates the behavior of economic indicators, their analysis and prediction. Financial market processes, which by their nature are nonlinear and non-stationary, are now undergoing a rigorous analysis. Therefore, to study such processes, it is necessary to use rejection models and methods for evaluating their parameters. A series of models describing nonlinear non-stationary processes are investigated in this paper.

## ЗМІСТ

<b>ПЕРЕЛІК СКОРОЧЕНЬ .....</b>	<b>9</b>
<b>ВСТУП .....</b>	<b>10</b>
1.1 Важливість прогнозування економічних показників.....	12
1.2 Огляд нелінійних процесів .....	13
1.3 Методи оцінювання параметрів нелінійних нестационарних процесів...	14
1.4 Висновки до розділу .....	16
<b>РОЗДІЛ 2 МАТЕМАТИЧНИЙ ОПИС МОДЕЛЕЙ, КРИТЕРІЇВ ТА ПРОГНОЗУ .....</b>	<b>17</b>
2.1 Стаціонарність ряду.....	17
2.2 Що таке інтегрованість ряду .....	18
2.3 Тест Діккі-Фуллара.....	20
2.5 Перевірка на гетероскедастичність.....	23
2.6 Критерії адекватності математичних моделей .....	27
2.7 Критерії якості прогнозу .....	30
2.8 Моделі інтегрованого процесу .....	33
2.9 Моделі гетероскедастичного процесу .....	35
Модель авторегресії з умовною гетероскедастичністю(АРУГ) .....	35
Узагальнена модель авторегресії з умовною гетероскедастичністю .....	37
2.10 Висновок до розділу .....	38
<b>РОЗДІЛ 3 РЕАЛІЗАЦІЯ ЗА ДОПОМОГОЮ EVIEWS ТА ВЛАСНОГО ПЗ.....</b>	<b>39</b>
3.1 Ряд «BAVA2» .....	39
3.1.1 Опис ряду.....	39

3.1.3 Кубічний тренд+авторегресія 4 порядку .....	43
3.1.4 Кубічний тренд+авторегресія 1 порядку .....	45
3.2 Ряд «FB».....	49
3.2.1 Опис ряду.....	49
3.2.2 Кубічний тренд+авторегресія 4 порядку .....	53
3.2.3 Кубічний тренд+авторегресія 1 порядку .....	55
3.2.4 Кубічний тренд+авторегресія 3 порядку .....	57
3.4 Власне ПЗ .....	60
3.4.1 Модель кубічного тренду .....	60
3.4.2 Модель АРІКС(4,1,0).....	62
3.4.3 Модель АРІКС(1,1,0).....	64
3.4.4 Модель АРІКС(3,1,0).....	65
3.4.5 Ряд «Золото».....	68
3.5 Висновки до розділу .....	72
4.1 Постановка задачі техніко-економічного аналізу .....	74
4.2 Обґрунтування функцій та параметрів програмного продукту .....	74
4.3. Економічний аналіз варіантів розробки програмного продукту .....	81
4.4 Висновки до розділу .....	85
ВИСНОВКИ .....	86
ЛІТЕРАТУРА.....	87
ДОДАТОК А.....	90
ДОДАТОК Б .....	100



## ПЕРЕЛІК СКОРОЧЕНЬ

АР – авторегресія

АРКС – авторегресія з ковзним середнім

ЕОМ - електронно-обчислювальна машина

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

ПП – програмний продукт

СКП – середньоквадратична похибка

СП – середня похибка

СПП – середня похибка в процентах

ФВА – функціонально-вартісний аналіз

## ВСТУП

З фінансовими процесами ми зустрічаємося на кожному кроці нашого повсякденного життя. Взагалі практично всі процеси в природі, чи то економічні, екологічні, соціальні, політичні являються нелінійними нестационарними процесами. Очевидно, що фінансові процеси, які є досить складними за своєю природою, також відносяться до цих процесів.

Для того щоб детально описати нелінійні нестационарні процеси потрібно використовувати відповідні моделі. На сьогоднішній день існує чимало математичних моделей, які можна використовувати в якості сприятливих для фінансових процесів. Найбільш поширеними є : авотрегресія порядку  $p$  ( $AP(p)$ ), авторегресія з ковзним середнім( $APKC(p,q)$ ), моделі, що описують процеси з трендом(інтегровані процеси), моделі, що описують гетероскедастичні процеси, тобто процеси зі змінною дисперсією( $APUG$ ,  $UAPUG$ ,  $EAPUG$ ) та інші. Всі перелічені моделі, оцінюються відповідними методами, яких, на щастя, також існує чимало. Найбільші відомі методи оцінювання математичних моделей: метод найменших квадратів(МНК), рекурсивний МНК(РМНК), метод максимальної правдоподібності(ММП) та метод Монте-Карло для ланцюгів Маркова. Завдяки ряду тестів для оцінювання адекватності моделі, ми маємо можливість вибрати модель зі спектру моделей-кандидатів, найбільш сприятливу для прогнозування параметрів моделі.

Якість прийнятих рішень прямо пропорційне якості прогнозування поведінки їх в майбутньому. Основною метою моделювання фінансових процесів є їх прогнозування, тобто інформація, щодо оцінених параметрів моделі в майбутньому. Головним завданням фінансових прогнозів є своєчасне

отримання знань, необхідних для вивчення та аналізу процесів, що відбудуться в майбутньому.

## **РОЗДІЛ 1 АКТУАЛЬНІСТЬ ДАНОЇ ТЕМИ, НЕОБХІДНІСТЬ У ПРОГНОЗУВАННІ ЕКОНОМІЧНИХ ПОКАЗНИКІВ**

### **1.1 Важливість прогнозування економічних показників**

Важко переоцінити, наскільки важливим є прогнозування в економіці. Для кожного підприємства чи держави загалом надзвичайно важливим є попередити негативні фактори, знизити ризики, зупинити погану тенденцію. Звичайно, в цьому назвичайно допомагає саме прогнозування. Знаючи наперед негативні наслідки, можна їх уникнути.

Коли підприємство чи держава приймає якісь рішення, дуже важливо добре оцінювати ризики, розуміти наслідки.

Що на мікрорівнях, що на глобальному рівні дуже важливо роль в фінансах відіграє саме прогнозування, особливо у високорозвинутих країнах. Коли недостатньо добре оцінені прогнози, то наслідками можуть бути величезні збитки, кризи й інші фінансові проблеми.

У природі й фінансові, й політичні, технічні тощо процеси зазвичай є нелінійними нестационарними. Щодня ми зустрічаємось із процесами такої природи, тому надзвичайно актуальним є прогнозування саме цих процесів. Щоб описати такі процеси, необхідно використати певні моделі, яких існує дуже багато. Ми розглянемо різні моделі інтегрованого процесу та гетероскадастичного. Також застосуємо різні методи оцінювання коефіцієнтів та прогнозування.

## 1.2 Огляд нелінійних процесів

На відміну від лінійних, нелінійні процеси не можна описати і відповідно оцінити звичайними лінійними методами. Нелінійні процеси також поділяються на стаціонарні та нестаціонарні. Прикладом нелінійних стаціонарних процесів є кусочно-стаціонарні процеси. Тобто на одному проміжку часу процес веде себе як стаціонарний, на іншому – як нестаціонарний.

Нелінійні нестаціонарні процеси в свою чергу поділяють на: інтегровані, коінтегровані та гетероскедастичні.

З інтегрованим процесом, він же процес із трендом або процес з одиничними коренями, ми зустрічаємося тоді, коли  $E[y(k)] \neq \text{const}$ , тобто математичне сподівання змінюється на часовому проміжку, який ми досліджуємо. Взагалі трендом називається поточне середнє значення процесу, яке вказує на його довгострокові зміни.

Інтегровані процеси можна розділити на процеси з детермінованим трендом та стохастичним трендом. Детермінованим називається тренд, який не містить випадкову складову. Коефіцієнти детермінованого тренду не змінюються протягом довгого проміжку часу. Вище розглянутий лінійний тренд (1.4) вважається детермінованим. Для перевірки процесу на наявність нелінійного детермінованого тренду можна оцінити рівняння:

$$y(k) = a_0 + c_1 k + c_2 k^2 + \dots + c_m k^m$$

Тобто модель у вигляді поліному порядку  $m$  відносно часу. Якщо хоча б один із коефіцієнтів  $c_i$ ,  $i = 1..m$  є статистично значимим, то гіпотеза щодо

відсутності тренду відхиляється. Якщо тренд містить випадкову компоненту, то він називається стохастичним трендом. Коефіцієнти такого тренду швидко змінюються в часі.

Гетероскедастичним процесом називають процес зі змінною дисперсією, тобто  $var[y(k)] = E\{[y(k) - \mu]^2\} \neq const$ . При розгляді фінансових процесів дисперсію розглядають як міру волатильності - показник, який вказує на коливання часових рядів. У фінансових процесах волатильність характеризує зміни цін в часі. Являється головним показником фінансових ризиків.

Яскравими представниками процесів із змінною в часі дисперсією, є: авто регресія умовно гетероскедастичністю (АРУГ), узагальнена авто регресія з умовною гетероскедастичністю (УАРУГ), експоненціальна УАРУГ (ЕУАРУГ) та інші. Детальніше про них в розділі 2.

### 1.3 Методи оцінювання параметрів нелінійних нестационарних процесів

На сьогоднішній день існує багато різноманітних методів оцінювання нестационарних процесів. Ми розглянемо найбільш поширені та найбільш практичні з них, а саме метод найменших квадратів(МНК) та рекурсивний МНК (РМНК).

#### Метод найменших квадратів(МНК)

Оцінки звичайного МНК обчислюють за допомогою виразу:

$$\hat{\theta} = [X^T X]^{-1} X^T y,$$

де  $\theta[p]$  – вектор оцінок параметрів розмірності  $p$ ;

$X[N \times p]$  – матриця вимірів незалежних змінних (регресорів);

$Y[N]$  – вектор залежної змінної.

Візьмемо для прикладу регресійну модель – множинну регресію:

$$y(k) = a_0 + a_1 x_1(k) + a_2 x_2(k) + a_3 x_3(k) + \varepsilon(k).$$

Матриця вимірів регресорів має вигляд

$$X = \begin{bmatrix} 1 & x_1(1) & x_2(1) & x_3(1) \\ 1 & x_1(2) & x_2(2) & x_3(2) \\ \cdot & \cdot & \cdot & \cdot \\ 1 & x_1(N) & x_2(N) & x_3(N) \end{bmatrix}$$

Одиниці першому стовпчику матриці  $X$  говорять про те, що вимір при коефіцієнті  $a_0$  завжди дорівнює одиниці.

Рекурсивний метод найменших квадратів (РМНК)

По своїй природі РМНК – ітеративна процедура оцінювання параметрів математичних моделей. Нехай є усталені початкові значення алгоритму  $\theta_0 =$

$\alpha, P_0 = \beta I, \beta \gg 0$ .  $i$ -тий крок рекурсивного алгоритму будемо шукати у вигляді:

$$P_i = P_{i-1} - \frac{P_{i-1} x^T(i) x(i) P_{i-1}}{1 + x(i) P_{i-1} x^T(i)},$$

$$\theta_i = \theta_{i-1} + P_i x^T(i) [y^T(i) - x(i) \theta_{i-1}]$$

де  $x(i)$  – це  $i$ -й рядок матриці вимірів.

#### 1.4 Висновки до розділу

Отже, практично всі процеси в природі являються нелінійними та нестационарними. Яскравими представниками таких процесів є нелінійні нестационарні процеси у фінансах, відповідно у даній роботі розглядається моделювання та прогнозування, на основі побудованих моделей таких процесів. На щастя, на сьогоднішній день існує багато моделей для оцінювання параметрів нелінійних нестационарних процесів та відповідних тестів на виявлення нестационарностей: наявності тренду чи гетероскедастичності. Таким чином, у розділі розглянуто найпоширеніші методи оцінювання параметрів моделей: метод найменших квадратів(МНК) та рекурсивний метод найменших квадратів(РМНК).



## РОЗДІЛ 2 МАТЕМАТИЧНИЙ ОПИС МОДЕЛЕЙ, КРИТЕРІЇВ ТА ПРОГНОЗУ

### 2.1 Стаціонарність ряду

Стаціонарний ряд - це ряд, статистичні властивості якого, такі як середнє, дисперсія і автокореляція, є постійними в часі. Отже, нестаціонарний ряд - це ряд, статистичні властивості якого змінюються в часі.

Нестаціонарні дані спочатку повинні бути перетворені в стаціонарні (наприклад, шляхом видалення тренду), щоб можна було провести подальший статистичний аналіз нетрендових стаціонарних даних. Це пов'язано з тим, що, наприклад, якщо ряд послідовно збільшується з плином часу, то середнє значення і дисперсія вибірки будуть збільшуватися разом з розміром вибірки, і вони завжди будуть занижувати середнє значення і дисперсію в майбутніх періодах. Звичайне середнє "позбавлення тренду" ряду полягає в тому, щоб підігнати його під лінію регресії, а потім відняти від вихідних даних.

Більшість методів статистичного прогнозування засновані на припущенні, що тимчасові ряди приблизно стаціонарні. Стаціонарний ряд відносно легко передбачити: ви просто прогнозуєте, що його статистичні властивості в майбутньому будуть такими ж, як і в минулому.

На рисинку 2.1 бачимо приклад ряду з трендом та ряду, який ми отримуємо після позбавлення його тренду.

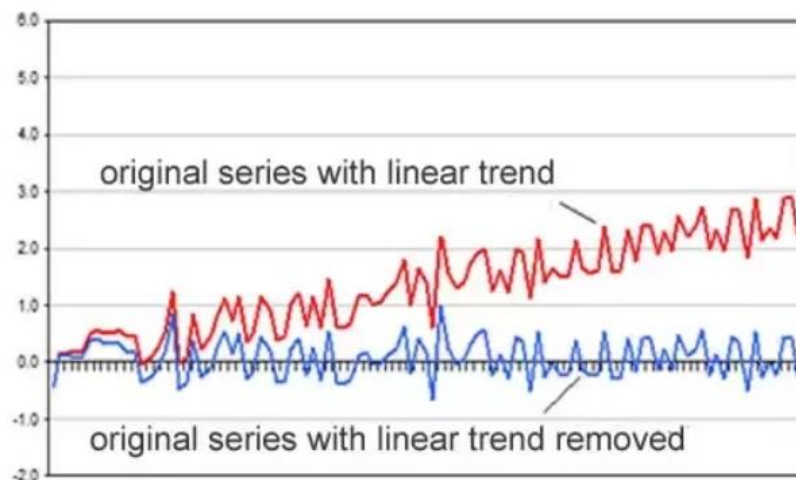


Рисунок 2.1 – Приклад ряду із трендом та цього ж ряду, позбавленого тренду

## 2.2 Що таке інтегрованість ряду

Якщо математичне очікування ряду змінюється протягом часу, тоді цей процес називають інтегрованим.

Середнє значення іноді спадає, іноді зростає. Ряди з трендом відповідно поділяються за цією ознакою. Також такі процеси розділяються залежно від детермінованості або стохастичності змін у часі.

Для опису таких процесів використовують різні моделі, які залежні від часу. Ми будемо використовувати поліноми від часу у вигляді:

$$y(K) = a_0 + a_1 \cdot K + a_2 \cdot K^2 + \dots + a_m \cdot K^m + \varepsilon(K),$$

де  $K$  – час, який дискретизував період реєстрації;

$\varepsilon(K)$  — похибки;

Коли на ряд накладемо дану модель, то похибки будуть включати в себе коливання.

Після накладання даної моделі на ряд, ми видаляємо тренд з неї. Надалі можемо розглядати залишки як модель без тренду.

Нестационарні часові ряди, які після видалення детермінованого тренду за допомогою МНК приводять ряд до стаціонарного ряду залишків, називаються стаціонарними відносно детермінованого тренду (trend stationary time). Якщо ж даний процес після диференціювання стає стаціонарним, то він називається стаціонарним відносно взяття різниць (difference stationary time series). В даній роботі ми розглянемо останній, тобто інтегрований.

Якщо після диференціювання порядку  $d$  процес стає стаціонарним, то початковий процес називається інтегрованим порядку  $d$ .

Тренд показує обстежену тенденцію збільшення або зміну даних в ході тривалого періоду часу. Тренд представляє собою сглажену, загальну, довгострокову, середню тенденцію. Не завжди потрібно, щоб зростання або спадання було в одному і тому ж напрямку протягом даного періоду часу.

Очевидно, що тренди можуть збільшуватись, змінюватись або бути стабільними у різних періодах часу. Загальний тренд має бути зростаючим, спадаючим або стабільною. Населення, сільськогосподарське виробництво, виробництво продукції, кількість народжених і смертність, кількість шкіл або коледжів - ось деякі з прикладів, що демонструють те чи інші тенденції руху.

### 2.3 Тест Діккі-Фуллара

1) враховано лише лагу першого порядку;

$$\Delta y(k) = b * y(k-1) + \varepsilon(k)$$

1) враховано зміщення;

$$\Delta y(k) = a_0 + b * y(k-1) + \varepsilon(k)$$

2) враховано тренд та зміщення;

$$\Delta y(k) = a_0 + a_1 * k + b * y(k-1) + \varepsilon(k)$$

де  $\varepsilon(k)$  - похибка;

$y(k)$  – наш процес;

$$\Delta y(k) = y(k) - y(k-1)$$

Ряд інтегрований, якщо у даному рівнянні  $b$  дорівнює нулю. Тоді значення  $p$  відповідно більше на один.

Якщо ж  $b$  приймає значення, що менше за нуль, то  $p$  відповідно теж буде менший за одиницю і ряди вважатиметься стаціонарним.

Вважатимемо за нуль-гіпотезу твердження, що ряд є нестаціонарним. Відповідно альтернативна гіпотеза справджується, якщо ряд є стаціонарним.

Для трьох версій даного тесту маємо відповідні критичні значення  $t$ -статистики.

$$|\tau| = \left| \frac{\hat{b}}{SE_{\hat{b}}} \right| > |\tau_{\text{крит}}|$$

В чисельнику оцінені коефіцієнти  $b$ , а у знаменнику стандартна похибка цієї оцінки. Якщо дана нерівність справджується, тоді нуль-гіпотеза відхиляється.

## 2.4 Що таке гетероскедастичність

Гетероскедастичність означає нерівномірне розсіювання. У регресійному аналізі мова йде про гетероскедастичність в контексті залишків або члена помилки. Зокрема, гетероскедастичність - це систематична зміна розкиду залишків в діапазоні вимірюваних значень. Гетероскедастичність є проблемою, так як регресія звичайних найменших квадратів (OLS) передбачає, що всі залишки взяті з сукупності, що має постійну дисперсію (гомоскедастичність).

Щоб задовільнити припущення про регресії і бути в змозі довіряти результатам, залишки повинні мати постійну дисперсію.

Взагалі кажучи, якщо ви бачите закономірності в залишках, то у вашої моделі виникає проблема, і ви можете не довіряти результатам.

Гетероскедастичність призводить до появи на графіках залишків виразною форми віяла або конуса. Щоб перевірити гетероскедастичність, необхідно оцінити залишки по спеціально підібраним ціннісним графіками. Зазвичай, при гетероскедастичності, зі збільшенням підігнаних значень, збільшується і дисперсія залишків.

Приклад такого малюнка конусоподібної форми в залишках можна подивитися на наведеному нижче рисунку 2.2. Зверніть увагу на те, як збільшується вертикальний діапазон залишків у міру збільшення встановлених значень.

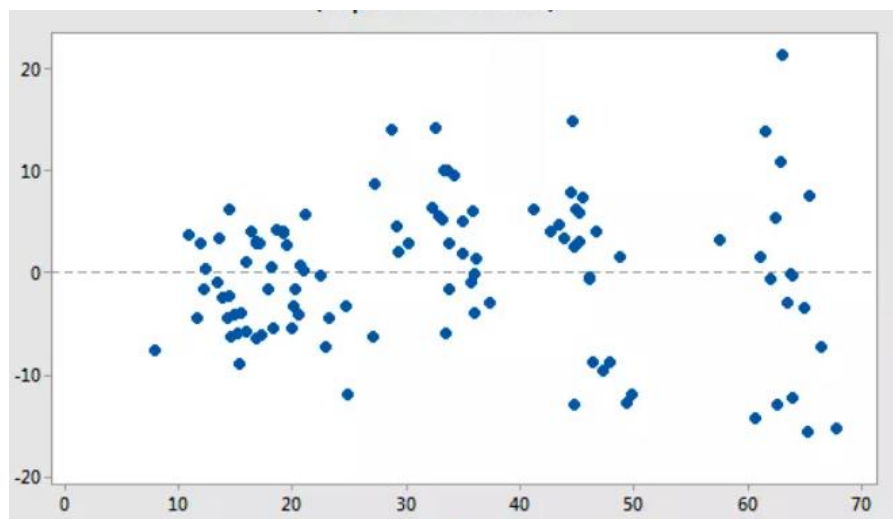


Рисунок 2.2 – Приклад гетероскедастичного процесу

Що викликає гетероскедастичність?

Гетероскедастичність частіше зустрічається в наборах даних, які мають великий діапазон між найбільшим і найменшим спостереженнями. Хоча існує безліч причин, за якими може існувати гетероскедастичність, загальне пояснення полягає в тому, що дисперсія помилки змінюється пропорційно фактору. Цей фактор може бути змінний в моделі.

У деяких випадках дисперсія збільшується пропорційно до цього фактору, але залишається постійною в процентах. Наприклад, 10-процентна зміна в такому числі, як 100, набагато менше 10-відсоткової зміни в такій великій кількості, як 100 000. У цьому сценарії очікуються більші залишки, пов'язані з більш високими значеннями. Тому потрібно бути обережним при роботі з великими діапазонами значень!

Оскільки великі діапазони пов'язані з цією проблемою, деякі типи моделей більш схильні до гетероскедастичності.

## 2.5 Перевірка на гетероскедастичність

Для перевірки ряду на гетероскедастичність можна застосувати різні тести, найпоширенішими з них є:

- 1) Бройша-Пагана
- 2) Уайта
- 3) Голдфельда-Квандта

Розглянемо кожен з них.

Бройша-Пагана

Даний тест ми застосовуємо в тому випадку, коли дисперсія випадкових похибок залежить від певної множини змінних.

Нехай маємо наступну модель:

$$y(k) = x^T(k)\beta + \varepsilon(k),$$

де  $x^T(k) = [1 \ x_2(k) \ x_3(k) \ \dots \ x_r(k)]$ .

Нехай гетероскедастичність матиме таку форму:

$$E[\varepsilon(k)] = 0, \quad \forall k,$$

$$\text{var}[\varepsilon(k)] = E[\varepsilon^2(k)] = \sigma_\varepsilon^2 = h(\alpha z^T(k)),$$

де  $z^T(k) = [1 \ z_2(k) \ z_3(k) \ \dots \ z_p(k)]$  - відомі змінні;

$\alpha = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_p]$  - невідомі коефіцієнти;

$h(\dots)$  - може приймати лише додатні значення.

Якщо всі коефіцієнти дорівнюють нулю, то нуль-гіпотеза підтверджується.

Потрібно оцінити параметри нашої моделі. Застосуємо МНК.

Отримуємо певні залишки моделі:

$$e(k) = y(k) - x^T(k)\beta$$

матимемо наступну дисперсію:

$$\sigma_\varepsilon^2 = N^{-1} \sum \varepsilon^2(k)$$

Знову використовуємо МНК та оцінюємо регресію  $\varepsilon^2(k)/\sigma_\varepsilon^2$ , потім знаходимо ESS:

$$\text{ESS} = \beta^T X^T X \beta - N\mu_y^2,$$

де  $\mu_y$  - середнє значення  $\{ y(k) \}$ .

Тест  $\text{ESS}/2$  розподіляється як  $\chi^2(p-1)$ .



Знаходимо критичні значення для  $\chi^2$ . Якщо отримана статистика більша за критичні значення, то гіпотеза відхиляється і ряд є нестационарним.

Уайта

Після того, як ми застосували один із методів оцінки коефіцієнтів, будується ще одна регресійна модель для квадратів залишків. Ця модель включатиме в себе регресори, квадрати регресорів, попарні добутки та константу.

Розглянемо наступне рівняння:

$$y(k) = a_0 + a_1 x_1(k) + a_2 x_2(k) + \varepsilon(k)$$

Регресори, попарні добутки та квадрати наступні:

$$[1 \ x_1 \ x_2 \ x_1^2 \ x_2^2 \ x_1 x_2]$$

Додаткова регресія буде незначущою.

Застосуємо LM-статистику.

$$LM = nR^2,$$

де  $n$  – число спостережень;

$R^2$ - коефіцієнт детермінації.

Якщо статистика буде більшою за критичне значення ( при відповідному рівні значущості), тоді нуль-гіпотеза про відсутність гетероскедастичності відхиляється.

Тест Голдфелда-Квандта

Якщо стандартне відхилення похибок пропорційне до однієї із змінних, тоді можна використати даний тест.

Почнемо із сортування змінної  $z$ , через яку ми допустили гетероскедастичність ряду.

Використовуємо МНК та оцінимо нашу обрану модель-кандидата для різних вибірок. Наші спостереження розділимо на дві рівні частини і вважатимемо за дві вибірки. Для обох вибірок знайдемо суми квадратів залишків та обрахуємо:

$$F = \frac{RSS_1/(q-k)}{RSS_2/(q-k)} = \frac{RSS_1}{RSS_2} .$$

Якщо ряд гомоскедастичний, тоді ця статистика матиме розподіл Фішера.

Аналогічно минулим тестам, якщо статистика матиме значення більше за критичне, яке визначається відповідно до рівня значущості, то гіпотеза про стаціонарність відхиляється, отже ряд буде вважатись гетероскедастичним.

## 2.6 Критерії адекватності математичних моделей

Звісно, деякі із методів прогнозування не дають достатньо хорошого результату. Потрібно покращувати ефективність прогнозу.

Звісно, побудована модель повинна бути достатньо адекватною, для цього і обраховуються відповідні критерії.

Для кожного процесу спочатку підбираємо моделі-кандидати, це рішення приймається на основі аналізу початкового стану цього процесу та відповідних літературних джерел. Робляться припущення щодо можливих моделей, які ймовірно добре описуватимуть даний ряд.

До кожного ряду індивідуальний підхід. Нечасто можна знайти для процесу уже готові моделі, частіше потрібно їх підбирати.

Потрібно підготувати дані у необхідній формі, потім оцінити коефіцієнти вище-описаних моделей-кандидатів. Потім вони оцінюються за допомогою критеріїв адекватності.

Розглянемо деякі із критеріїв адекватності:

### 1) **t-Statistic** (t-статистика Стьюдента)

Для її обчислення потрібно поділити коефіцієнт на стандартну похибку. Таким чином ми оцінюємо гіпотезу на рівність нулю.

Критерій винайшли, щоб оцінювати якість пива Гіннес, але оскільки не можна було розповідати секрети компанії, тому Госсет випустив дану інформацію у газеті під псевдонімом Стьюдент.

$$t_a = \frac{\hat{a} - a_0}{SE_{\hat{a}}},$$

де  $a_0$  – нульова гіпотеза;

$\hat{a}$  – оцінка коефіцієнта, яку видає пакет програми;

$SE_{\hat{a}}$  – стандартна похибка (також обраховується пакетом);

Із формули можна побачити, що при меншій похибці, маємо кращу оцінку для моделі.

Важливим для оцінки значимості коефіцієнта є рівень значимості  $\alpha$ .

Його приймають рівним 10%, 5% або 1%. Якщо він рівний, наприклад, 1%, то це значить, що ми допускаємо помилку при прийнятті рішення у 1% випадків. Тоді потрібно взяти відповідні таблиці критичних значень і порівняти із отриманим.

Якщо за модулем отримане значення буде меншим від критичного, то приймається нуль-гіпотеза і коефіцієнт вважається незначимим.

Якщо за модулем отримане значення більше, ніж критичне, то коефіцієнт вважається значимим.

Для обрахунку цього значення потрібно поділити дисперсію залежної змінної, яку ми отримати за допомогою нашої моделі-кандидата, на дисперсію, отриманої на основі фактичних значень.

Якщо модель хороша, то  $R^2 \rightarrow 1$ .

$$R^2 = \frac{var(\hat{y})}{var(y)},$$

де  $var(\hat{y})$  дисперсія залежної змінної, яку ми отримати за допомогою нашої моделі-кандидата;

$var(y)$  - дисперсію, отриманої на основі фактичних значень.

Таким чином ми знаходимо значення інформованості нашої моделі відносно інформованості наших даних.

## 2) **Sum square resid** (Сума квадратів похибок моделі ) SSE

Для обчислення даного критерія потрібно скористатись наступною формулою:

$$SSE = \sum_{t=1}^N (y - \hat{y})^2 \quad ,$$

де  $y$  — справжні значення;

$\hat{y}$  — значення, отримані після побудови нашої моделі;

$N$  — розмір нашої вибірки;

Звісно, найкращим кандидатом вважатиметься той, що набуває найменшого значення.

## 3) **Akaike info criterion** (Інформаційний критерій Акайке (AIC)).

Скористаємось формулою:

$$AIC = -\frac{2 \cdot l}{T} + \frac{2 \cdot k}{T},$$

де  $l$  – значення логарифмічної функції правдоподібності.

Чим менше значення набуває даний критерій, тим адекватнішою вважається модель  $AIC \rightarrow \min$ .

4) **Durbin-Watson stat** (Коефіцієнт Дурбіна-Уотсона (DW)).

$$DW = \frac{\sum_{t=2}^N (\varepsilon_t - \varepsilon_{t-1})^2}{\sum_{t=1}^N \varepsilon_t^2},$$

де  $\varepsilon$  – вектор залишків, а саме різниця між справжніми та отриманими значеннями.

Для гарної моделі даний коефіцієнт близький до 2. В такому випадку можна зробити висновок, що автокореляції залишків немає.

5) Коефіцієнт Тейла

$$U = \frac{\sqrt{\frac{1}{N} \sum_{k=1}^N [y(k) - \hat{y}(k)]^2}}{\sqrt{\frac{1}{N} \sum_{k=1}^N y^2(k) + \frac{1}{N} \sum_{k=1}^N \hat{y}^2(k)}}$$

Це критерій точності моделі. Значення знаходяться між 0 та 1. Якщо коефіцієнт близький до 1, тоді обраний кандидат-модель не підходить.

## 2.7 Критерії якості прогнозу

Основним завданням даної роботи є прогноз. Тому надзвичайно важливо порівняти прогноз при кожній моделі. Для цього і потрібні критерії якості прогнозу.

Основними критеріями є:

1) **S.E. of regression** середньоквадратична похибка (СКП)

$$СКП(y, \hat{y}) = RMSE(y, \hat{y}) = \sqrt{MSE(y, \hat{y})} = \sqrt{E((y - \hat{y})^2)} = \sqrt{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}},$$

де  $\hat{y}$  – значення  $y$ , отримані за допомогою нашої моделі;

$y$  – справжні значення;

$N$  – розмірність;

Звісно, модель є найкращою при найменшому значенні цього критерія.

2) Сума похибок (СП)

Розрахувати даний показник можна за формулою:

$$СП = \frac{1}{N} \sum_{k=1}^N [y(k) - \hat{y}(k)] = \frac{1}{N} \sum_{k=1}^N e(k),$$

де  $\hat{y}$  – значення  $y$ , отримані за допомогою нашої моделі;

$y$  – справжні значення;

$N$  – розмірність;

Найкращим є прогноз, який дасть значення  $СП=0$ .

### 3) Середня похибка у процентах (СПП)

$$СПП = \frac{1}{N} \sum_{k=1}^N \frac{[y(k) - \hat{y}(k)]}{y(k)} \times 100\%,$$

де  $\hat{y}$  – значення  $y$ , отримані за допомогою нашої моделі;

$y$  – справжні значення;

$N$  – розмірність;

При СПП меншому за 5% можна вважати, що прогноз є досить ефективним. Ідеальним значенням, звичайно, також вважається нуль.

### 4) Середнє абсолютне значення похибки в процентах (САПП)

$$САПП = \frac{1}{N} \sum_{k=1}^N \frac{|y(k) - \hat{y}(k)|}{|y(k)|} \times 100\%.$$

де  $\hat{y}$  – значення  $y$ , отримані за допомогою нашої моделі;

$y$  – справжні значення;

$N$  – розмірність;

### 5) Максимальна абсолютна похибка (МАП).



$$\text{МАП} = \max_k \{|y(k) - \hat{y}(k)|\}, 1 \leq k \leq N,$$

де  $\hat{y}$  – значення  $y$ , отримані за допомогою нашої моделі;

$y$  – справжні значення;

$N$  – розмірність;

б) Мінімальна абсолютна похибка (МіАП)

$$\text{МіАП} = \min_k \{|y(k) - \hat{y}(k)|\}, 1 \leq k \leq N,$$

де  $\hat{y}$  – значення  $y$ , отримані за допомогою нашої моделі;

$y$  – справжні значення;

$N$  – розмірність;

Кожен критерій є досить точним та важливим. Бажано використовувати відразу декілька з них.

## 2.8 Моделі інтегрованого процесу

### Поліноміальна модель

Поліноміальна модель представляє собою рівняння:

$$y(k) = c_0 + c_1 k + c_2 k^2 + \dots + c_m k^m + \varepsilon(k)$$

де  $k$  – дискретний час, що має прямий зв'язок з неперервним часом  $t$  через період дискретизації  $t = kT_s$  ;

$\varepsilon(k)$  – випадкова змінна, введення якої у модель процесу зумовлено кількома причинами: помилки у структурі моделі; методичні похибки оцінювання параметрів моделей; вплив зовнішніх випадкових збурень; вплив похибок вимірів. Її оцінку знаходимо, оцінюючи рівняння:

$$\hat{\varepsilon}(k) = e(k),$$

де  $e(k)$  – похибка моделі – містить коливання, які накладаються на тренд.

По суті, опис тренду рівнянням видаляє його з нашого процесу, тобто повна модель буде складатися щонайменше з: рівняння для тренду і рівняння авторегресії з ковзним середнім АРКС(p,q):

$$y(k) = a_0 + \sum_{i=1}^p a_i(k-i) + \sum_{j=1}^q b_j(k-j) + \varepsilon(k),$$

що описує коливання, які накладаються на тренд.

Одним з найпоширеніших способів видалення тренду є видалення за допомогою різниць. Відповідно перші різниці видаляють тренд першого порядку, другі – квадратичний тренд і так далі. Візьмемо для прикладу лінійний тренд  $y(k) = a_0 + a_1 k$ . Перші різниці будуть мати вигляд:

$$\begin{aligned} \Delta y(k) &= y(k) - y(k-1) = a_0 + a_1 k - [a_0 + a_1(k-1)] = \\ &= a_0 + a_1 k - a_0 - a_1 k + a_1 = a_1, \end{aligned}$$

тобто ми бачимо фактичне видалення тренду.

Ще раз наголосимо на тому, що випадкову змінну  $\varepsilon(k)$  шукаємо по залишках, відповідно оцінивши параметри рівняння  $y(k) = c_0 + c_1 k + c_2 k^2 + \dots + c_m k^m$ . Таке рівняння легко оцінюється звичайним методом найменших квадратів, наприклад, за допомогою пакету статистичної обробки даних EViews (про нього далі).

## 2.9 Моделі гетероскедастичного процесу

Безумовно найпоширенішими моделями гетероскедастичних процесів є модель авторегресії з умовною гетероскедастичністю (АРУГ) та узагальнена АРУГ(УАРУГ). Розберемо детальніше кожну з них.

### Модель авторегресії з умовною гетероскедастичністю(АРУГ)

Припустимо, що ми маємо модель:

$$y(k) = a_0 + a_1 y(k-1) + \varepsilon(k).$$

Позначимо через  $\{\hat{\varepsilon}(k)\}$  – оцінки залишків моделі. Відповідно умовну дисперсію визначаємо за формулою:

$$\text{Var}[y(k+1) | y(k)] = E_k\{[y(k+1) - a_0 - a_1 y(k)]^2\} = E_k[\hat{\varepsilon}^2(k+1)].$$

Застосуємо модель авторегресії(АР(q)) для квадратів оцінок залишків моделі:

$$\hat{\varepsilon}^2(k) = \alpha_0 + \alpha_1 \hat{\varepsilon}^2(k-1) + \alpha_2 \hat{\varepsilon}^2(k-2) + \dots + \alpha_q \hat{\varepsilon}^2(k-q) + v(k),$$

де  $v(k)$  – гаусівський процес (білого шуму) нульове середня для адекватної моделі якого дорівнює нулю.

При випадку, коли  $\alpha_2 = \alpha_3 = \dots = \alpha_q = 0$ , оцінка дисперсії прийме значення константи. Рівняння (2.10) може допомогти у разі однокрокового прогнозування умовної дисперсії:

$$E_k[\hat{\varepsilon}^2(k+1)] = \alpha_0 + \alpha_1 \hat{\varepsilon}^2(k) + \alpha_2 \hat{\varepsilon}^2(k-1) + \dots + \alpha_q \hat{\varepsilon}^2(k-q+1).$$

Так як ми бачимо, що ця модель будується при умові нестационарності  $\text{Var}[y(k)] \neq \text{const}$ , то її називають авторегресійною з умовною гетероскедастичністю (АРУГ). Залишки  $\varepsilon(k)$  генеруються на основі регресійних, авторегресійних(АР) чи авторегресійних рівнянь з ковзним середнім(АРКС).

Модель АРУГ широко використовують для прогнозування волатильності. За її використання ми маємо більшу ймовірність отримати значення які дуже відрізняються від математичного сподівання. Основним недоліком таких моделей є те, що вона накладає обмеження на параметри, а це досить таки ускладнює їх оцінювання, тим паче, коли ми маємо справу з моделями великого порядку.

## Узагальнена модель авторегресії з умовною гетероскедастичністю (УАРУГ)

Модель УАРУГ базується на введенні в рівняння для оцінки умовної варіації компонентів ковзного середнього. В порядку речей, що на будь-якому фінансовому ринку постійно змінюється ціна. Ці зміни зазвичай описують рівнянням:

$$\varepsilon(k) = v(k)[h(k)]^{1/2}$$

Тоді розподіл умовної дисперсії описуватиметься моделлю:

$$h(k) = \alpha_0 + \sum_{i=1}^q \alpha_i \varepsilon^2(k-i) + \sum_{j=1}^p \beta_j h(k-j).$$

Так як процес  $v(k)$  визначений як гаусівський(процес білого шуму) і відповідно некорелює із значенням  $\varepsilon(k-1)$ , то як умовне так і безумовне середнє для  $\varepsilon(k)$  будуть дорівнювати нулю. Не складно здогадатися, що безумовне математичне очікування матиме вигляд:

$$E[\varepsilon(k)] = E[v(k)(h(k))^{1/2}] = 0$$

Звідси, умовна дисперсія  $\varepsilon(k)$  визначається як  $E_{k-1}[\varepsilon^2(k)] = h(k)$

Як ми бачимо рівняння узагальненої моделі авто регресії АР(p) з умовною гетероскедастичністю порядку q – УАРУГ(p,q), розбивається на дві

частини – авторегресія та ковзне середнє відносно змінної дисперсії, тобто дисперсії гетероскедастичного процесу. Якщо  $p = 0$ ,  $q = 1$ , то маємо справу з процесом першого порядку УАРУГ(0,1). Коли коефіцієнти  $\beta_i = 0$ , то модель УАРУГ(p,q) можна називати моделлю АРУГ(p,q). Для того щоб умовна дисперсія була скінченою, корені характеристичного рівняння повинні бути всередині одиничного кола.

Головна перевага УАРУГ моделі в тому, що вона дає нам можливість зменшити порядок моделі АРУГ. Також за допомогою цієї моделі ми можемо врахувати всі важкі розподіли випадкової величини.

## 2.10 Висновок до розділу

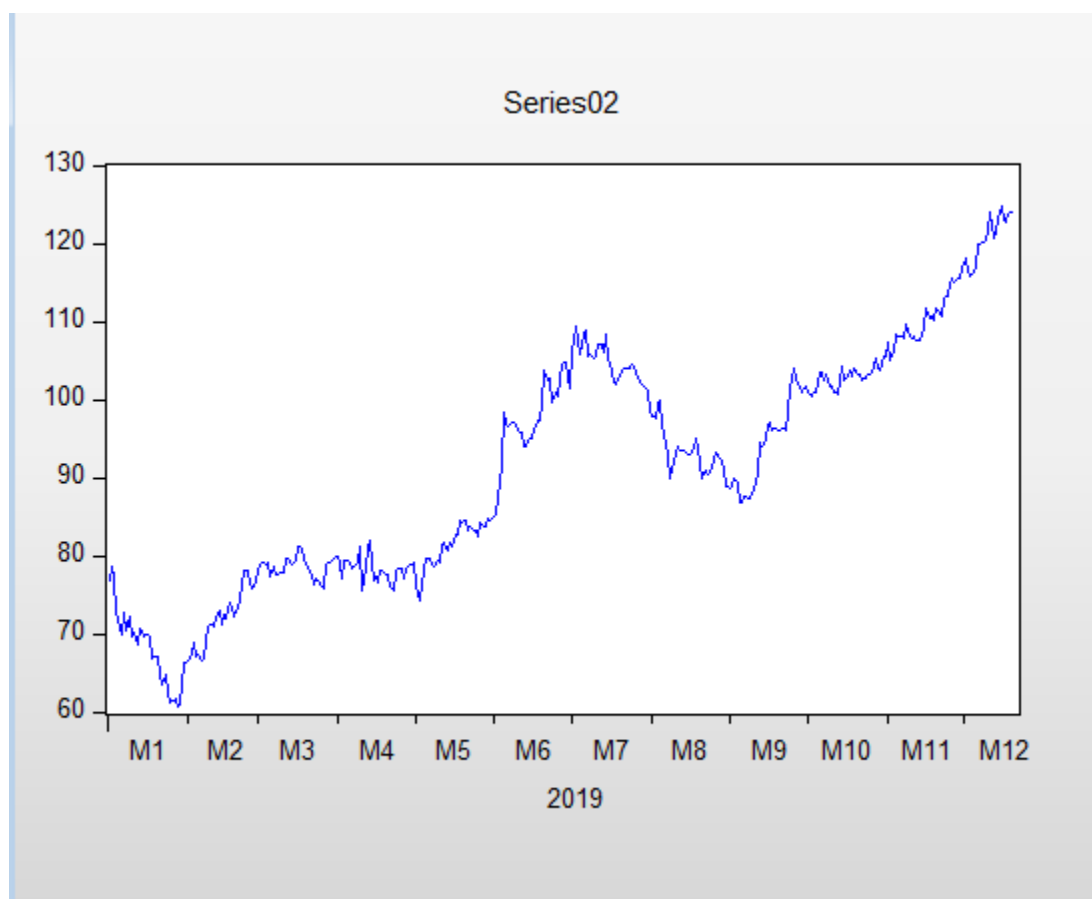
Отже, ми розглянули що таке інтегрований та гетероскедастичний ряди, як перевірити ряд на інтегованність та гетероскедастичність. Якщо виявлено, що ряд із трендом, то потрібно будувати моделі АРІКС. Якщо виявлено, що ряд гетероскедастичний, тоді будуємо моделі АРУГ та УАРУГ. Також розглянули критерії адекватності моделі: t-статистика Стьюдента, коефіцієнт детермінації, сума квадратів похибок моделі, інформаційний критерій Акайке, коефіцієнт Дурбина-Уотсона, коефіцієнт Тейла. Продемонстровано критерії оцінки прогнозів: середньоквадратична похибка, сума похибок, середня похибка у процентах та середнє абсолютне значення похибок у процентах. Кожен із критеріїв є достатньо точним і дуже важливим, але бажано використовувати відразу декілька з них.

## РОЗДІЛ 3 РЕАЛІЗАЦІЯ ЗА ДОПОМОГОЮ EVIEWS ТА ВЛАСНОГО ПЗ

### 3.1 Ряд «ВАВА2»

#### 3.1.1 Опис ряду

На рисунку 3.1 бачимо графік даного ряду, побудований в Eviews.



### Рисунок 3.1 - Графічне зображення часового ряду «ВАВА2»

Опис характеристик даного ряду:

Mean (середнє) – 91.12133;

Median (медіана) – 92.39000;

Maximum (максимум) – 124.7500;

Skewness (асиметрія) – 0.109398

Kurtosis (ексцес) – 2.069726

Jarque-Bera (статистика Жака-Бера) – 13.47091

Середнє значення не дорівнює нулю.

Ряд нестационарний, візуально видно, що присутній тренд.

Видалення лінійного тренда не приводить ряд до стаціонарного. Даний ряд не є TS-рядом.

Примінімо тест Дікі-Фуллера, щоб визначити, чи цей ряд інтегрований.

Null Hypothesis: Y has a unit root  
Exogenous: Constant, Linear Trend  
Lag Length: 0 (Automatic - based on SIC, maxlag=16)

	t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic	-2.306790	0.4287
Test critical values: 1% level	-3.984269	
5% level	-3.422606	
10% level	-3.134184	

\*MacKinnon (1996) one-sided p-values.



### Рисунок 3.2 - Примінення теста Дікі-Фуллера в Eviews

Оскільки  $\text{Prob} > 0.05$ , то гіпотеза про DS-нестационарність ряду приймається.

Визначимо порядок інтегрування даного ряду:

Якщо видалити різницю першого порядку, то маємо  $\text{Prob} < 0.05$ . Отже, гіпотеза про DS-нестационарність ряду  $d(y)$  відхиляється.

Тобто операція різниці першого порядку привела нестационарний ряд  $y$  до стаціонарного.

Отже, початковий ряд відноситься до процесів  $I(1)$ . Але після видалення тренду він переходить в  $I(0)$ .

#### 3.1.2 Модель кубічного тренду

$$y = c(1) + c(2)*k + c(3)*k*k + c(4)*k*k*k$$

За допомогою МНК Eviews оцінює коефіцієнти, маємо:

$$C(1) = 62.98248$$

$$C(2) = 0.264424$$

$$C(3) = -0.000893$$

$$C(4) = 1.68E-06$$

Для даної моделі маємо такі значення критеріїв адекватності:

$$R^2 = 0.840953;$$

$$\sum e^2 = 13284.14;$$

$$DW = 0.060223;$$

$$AIC = 6.485505$$

Отже, модель для тренду адекватна, але критерії загалом нас не задовільняють. Потрібно додати АРКС для залишків.

Будуємо прогноз на проміжок з 10/01/2019 по 12/20/2019 на основі значень даного ряду, графік бачимо на рисунку 3.3.



Рисунок 3.3 - Графічне зображення часового ряду «ВАВА2» та результату прогнозу на основі моделі кубічний тренд

СКП = 2.527; САПП = 2.2657%; Коефіцієнт Тейла = 2.3721 (2)

0.0135

Дану корелограму спостерігаємо на рисунку 3.4.

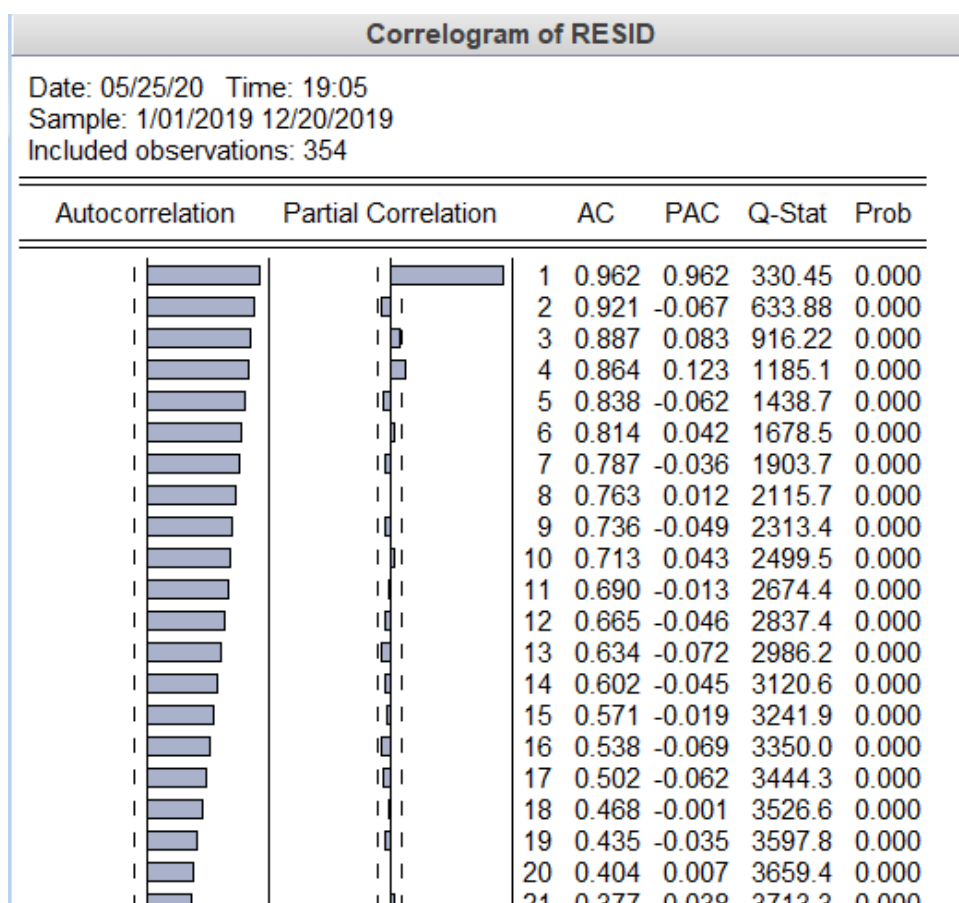


Рисунок 3.4 - Корелограма залишків

### 3.1.3 Кубічний тренд+авторегрсія 4 порядку

Будуємо для залишків наступну модель:

$$\text{res}(k)=c(1)+c(2)*\text{res}(k-1)+c(3)*\text{res}(k-2)+c(4)*\text{res}(k-3)+c(5)*\text{res}(k-4)+e(k)$$

Отже, це модель авторегресії четвертого порядку.

$$C(1) = -0.182541$$

$$C(2) = 0.976894$$

$$C(3) = -0.161749$$

$$C(4) = 0.129578$$

$$C(5) = 0.001148$$

Для даної моделі маємо такі значення критеріїв адекватності:

$$R^2 = 0.931307;$$

$$\sum e^2 = 424.2750;$$

$$DW = 1.975052;$$

$$AIC = 3.780616$$

Бачимо, що критерії адекватності стали приймати кращі значення.

Для прогнозу побудуємо в Eviews загальну модель авторегресії 4 порядку із кубічним трендом:

$$Y=c(1)+c(2)*y(-1)+c(3)*y(-2)+c(4)*y(-3)+c(5)*y(-4)+c(6)*k+c(6)*k^2+c(7)*k^3$$

Будуємо прогноз на проміжок з 10/01/2019 по 12/20/2019 на основі значень даного ряду, графік спостерігаємо на рисунку 3.5.

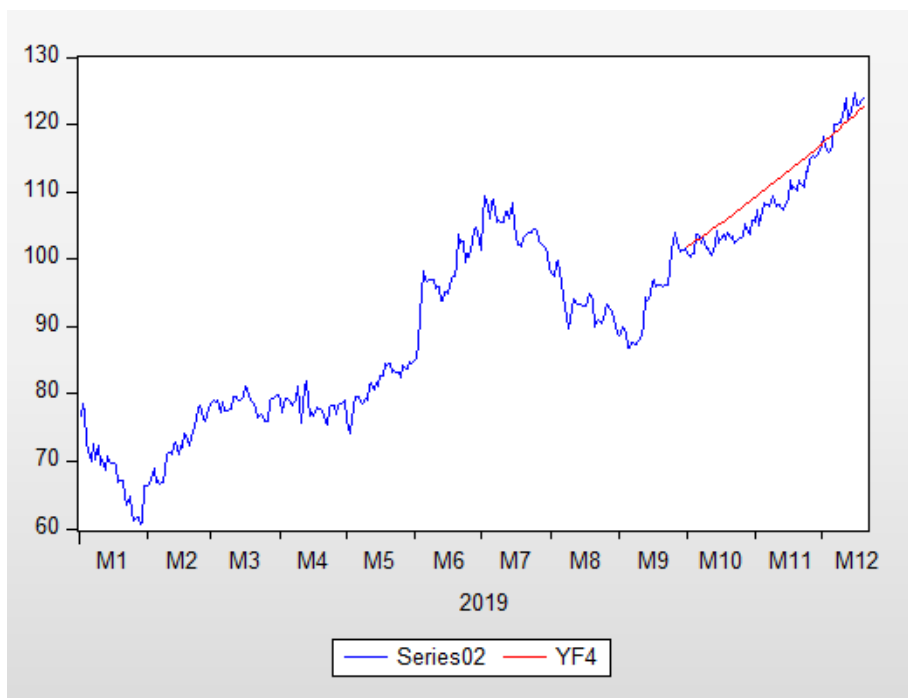


Рисунок 3.5 - Графічне зображення часового ряду «ВАВА2» та результату прогнозу на основі моделі  $AR(4)$ +кубічний тренд

СКП = 2.36; САПП = 2.185840%; Коефіцієнт Тейла = 2.28 (2)

0.012154

### 3.1.4 Кубічний тренд+авторегрсія 1 порядку

$$\text{res}(k) = c(1) + c(2) * \text{res}(k-1)$$

$$C(1) = -0.162972$$

$$C(2) = 0.942570$$

Для даної моделі маємо такі значення критеріїв адекватності:

$$R^2 = 0.929919;$$

$$\sum e^2 = 454.6827;$$

$$DW = 2.277689;$$

$$AIC = 3.02332;$$

Як бачимо, критерії свідчать, що дана модель є більш адекватною, ніж просто кубічного тренду, але дещо гірша, ніж кубічний тренд + авторегресія четвертого порядку.

Для прогнозу побудуємо в Eviews загальну модель авторегресії 1 порядку із кубічним трендом:

$$Y = c(1) + c(2) * y(-1) + c(3) * k + c(4) * k^2 + c(5) * k^3$$

Будуємо прогноз на проміжок з 10/01/2019 по 12/20/2019 на основі значень даного ряду, графік прогнозу на рисунку 3.6.

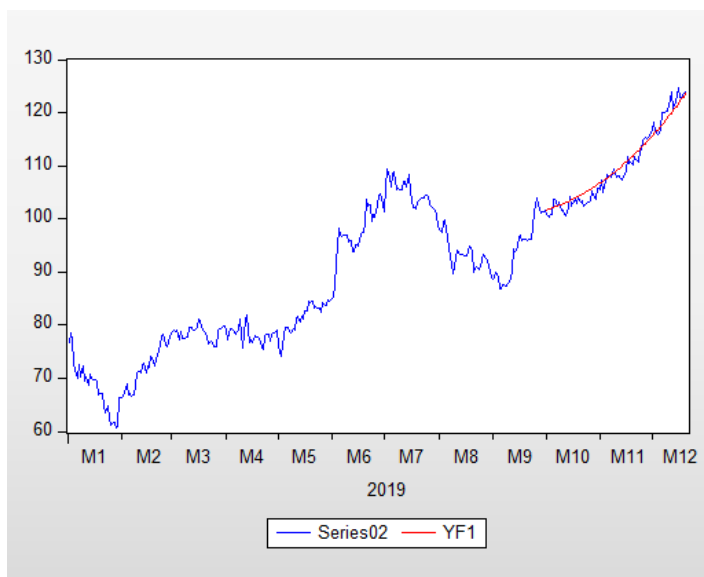


Рисунок 3.6 - Графічне зображення часового ряду «BAVA2» та результату прогнозу на основі моделі  $AR(1)$ +кубічний тренд

СКП = 1.21; САПП = 1.1%; Коефіцієнт Тейла = 1.19 (2)

0.006577

### 3.1.5 Кубічний тренд+авторегресія 3 порядку

$$\text{res}(k) = c(1) + c(2) * \text{res}(k-1) + c(3) * \text{res}(k-2) + c(4) * \text{res}(k-3)$$

$$C(1) = -0.209231$$

$$C(2) = 0.995915$$

$$C(3) = -0.203127$$

$$C(4) = 0.144776$$

Для даної моделі маємо такі значення критеріїв адекватності:

$$R^2 = 0.930126$$

$$\sum e^2 = 437.0856;$$

$$DW = 1.884270;$$

$$AIC = 3.792977;$$

Для прогнозу побудуємо в Eviews загальну модель авторегресії 1 порядку із кубічним трендом:

$$Y = c(1) + c(2) * y(-1) + c(3) * y(-2) + c(4) * y(-3) + c(4) * k + c(5) * k^2 + c(6) * k^3$$

Будуємо прогноз на проміжок з 10/01/2019 по 12/20/2019 на основі значень даного ряду.

Результати даного прогнозу спостерігаємо на рисунку 3.7.



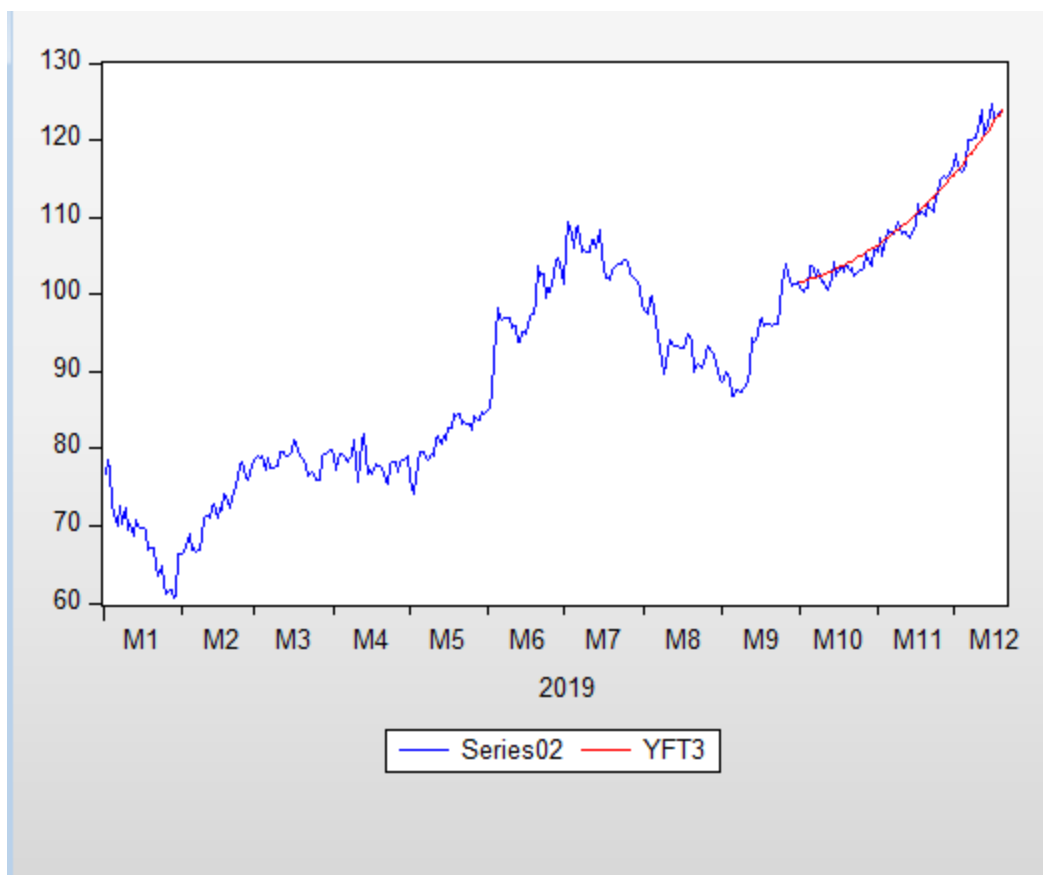


Рисунок 3.7 - Графічне зображення часового ряду «ВАВА2» та результату прогнозу на основі моделі  $AR(3)$ +кубічний тренд

СКП = 1.09; САПП = 0.99%; Коефіцієнт Тейла = 1.08 (2)

0.99322

### 3.2 Ряд «FB»

#### 3.2.1 Опис ряду

Графік даного ряду бачимо на рисунку 3.8.

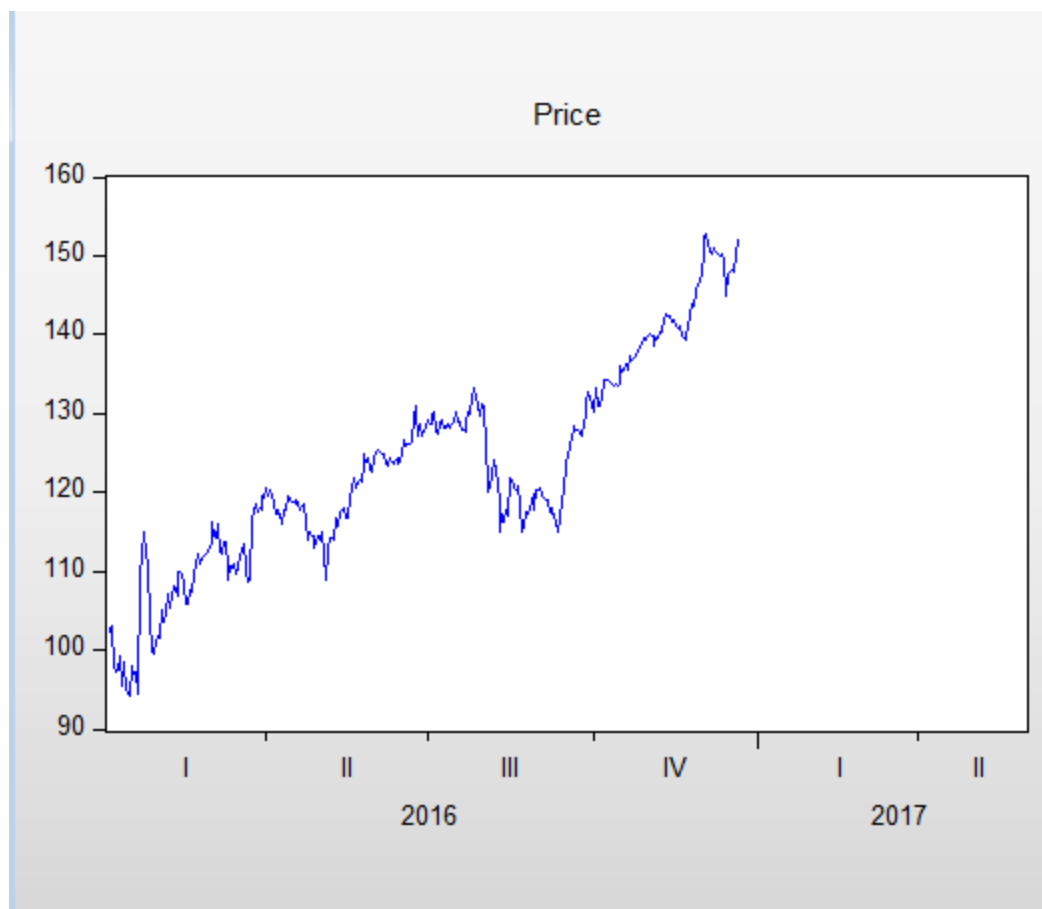


Рисунок 3.8 - Графічне зображення часового ряду «FB»

Опис характеристик даного ряду:

Mean (середнє) – 123.2995;

Median (медіана) – 121.1100;

Maximum (максимум)- 152.7800;

Skewness (асиметрія) – 0.204868;

Kurtosis (ексцес) – 2.670889;

Jarque-Bera (статистика Жака-Бера) – 4.073909;

Середнє значення не дорівнює нулю.

Примінімо тест Дікі-Фуллера, щоб визначити, чи цей ряд інтегрований.

Застосування даного тесту бачимо на рисунку 3.9.

Augmented Dickey-Fuller Unit Root Test on PRICE		
Null Hypothesis: PRICE has a unit root		
Exogenous: Constant, Linear Trend		
Lag Length: 0 (Automatic - based on SIC, maxlag=16)		
	t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic	-2.689510	0.2416
Test critical values: 1% level	-3.984269	
5% level	-3.422606	
10% level	-3.134184	

Рисунок 3.9 – Тест Дікі-Фуллера для часового ряду «FB»

Оскільки  $\text{Prob} > 0.05$ , то гіпотеза про DS-нестационарність ряду приймається.

#### 1. Модель кубічного тренду

$$y = c(1) + c(2)*k + c(3)*k*k + c(4)*k*k*k$$

За допомогою МНК Eviews оцінює коефіцієнти, маємо:

$$C(1) = 95.12943$$

$$C(2) = 0.400559$$

$$C(3) = -0.002128$$

$$C(4) = 4.16E-06$$

Для даної моделі маємо такі значення критеріїв адекватності:

$$R^2 = 0.884211;$$

$$\sum e^2 = 7097.334$$

$$DW = 0.15048;$$

$$AIC = 5.858653;$$

Будуємо прогноз (рисунок 3.10) на проміжок з 10/04/2016 по 5/30/2017 на основі значень даного ряду.

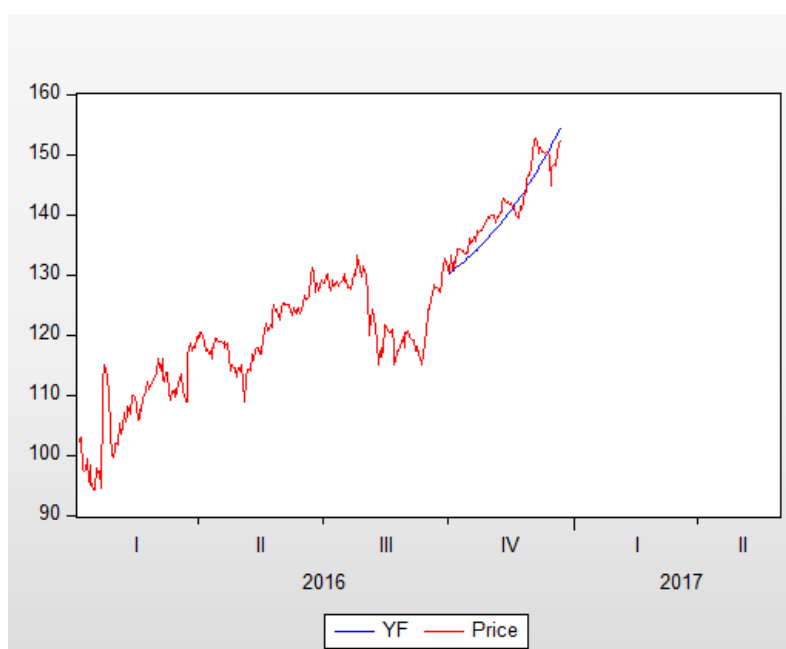


Рисунок 3.10 - Графічне зображення результату прогнозу на основі моделі кубічного тренду та часового ряду «FB»

$$СКП = 2.287714; \text{ САПП} = 1.603967\%; \text{ Коефіцієнт Тейла} = 2.3658 \quad (2)$$

$$0.009342$$

Корелограму демонструємо на рисунку 3.11.

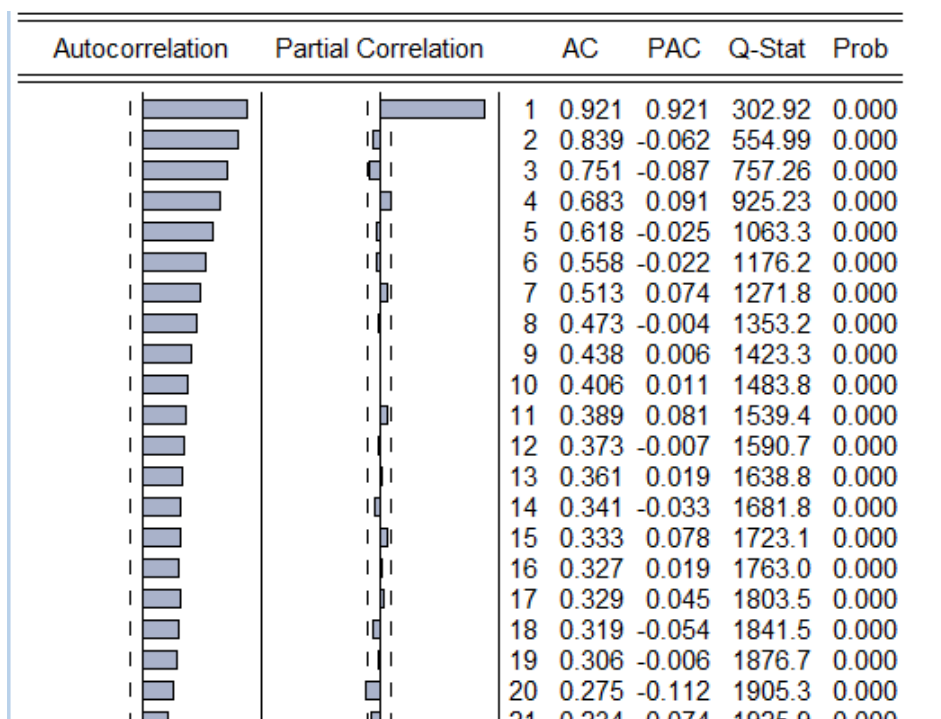


Рисунок 3.11 - Корелограма залишків

### 3.2.2 Кубічний тренд+авторегресія 4 порядку

Будуємо для залишків наступну модель:

$$\text{res}(k) = c(1) + c(2) * \text{res}(k-1) + c(3) * \text{res}(k-2) + c(4) * \text{res}(k-3) + c(5) * \text{res}(k-4) + e(k)$$

Отже, це модель авторегресії четвертого порядку.

$$C(1) = -0.067327$$

$$C(2) = 0.958288$$

$$C(3) = 0.115245$$

$$C(4) = -0.289842$$

$$C(5) = 0.120229$$

Для даної моделі маємо такі значення критеріїв адекватності:

$$R^2 = 0.841429;$$

$$\sum e^2 = 553.5622;$$

$$DW = 1.974626;$$

$$AIC = 4.046608;$$

Для прогнозу побудуємо в Eviews загальну модель авторегресії порядку із кубічним трендом:

$$Y = c(1) + c(2)*y(-1) + c(3)*y(-2) + c(4)*y(-3) + c(5)*y(-4) + c(6)*k + c(6)*k^2 + c(7)*k^3$$

Будуємо прогноз на проміжок з 10/04/2016 по 5/30/2017 на основі значень даного ряду.

$$СКП = 1.7499; \text{ САПП} = 1.211579\%; \text{ Коефіцієнт Тейла} = 1.96955$$

$$0.007946$$

Результати прогнозу бачимо на рисунку 3.12.

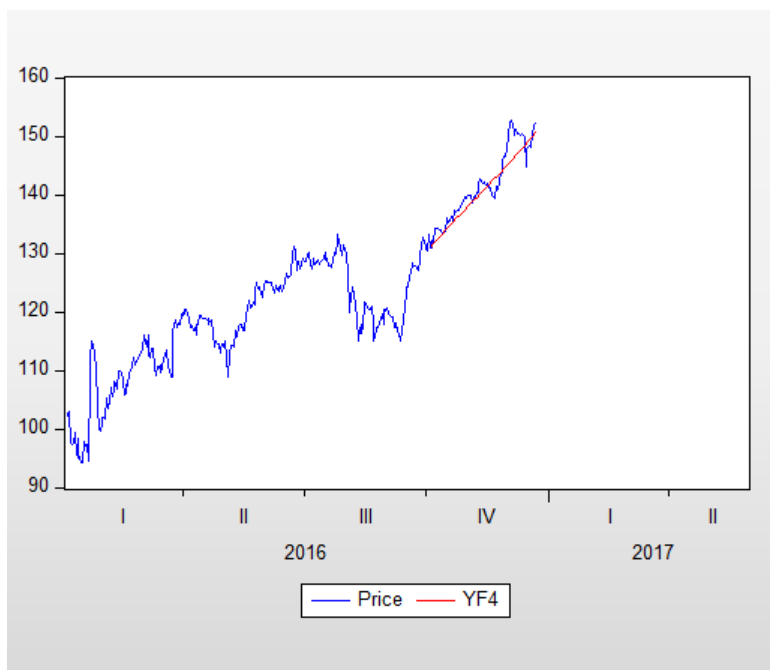


Рисунок 3.12 - Графічне зображення часового ряду «FB» та результату прогнозу на основі моделі  $AR(4)$ +кубічний тренд

### 3.2.3 Кубічний тренд+авторегресія 1 порядку

Авторегресія першого порядку  $AR(1)$ :

$$res(k) = c(1) + c(2) * res(k-1)$$

$$C(1) = -0.114083;$$

$$C(2) = 0.908959;$$

Для даної моделі маємо такі значення критеріїв адекватності:

$$R^2 = 0.829900;$$

$$\sum e^2 = 602.1759;$$

$$DW = 1.695213;$$

$AIC=4.084876$ ;

Для прогнозу побудуємо в Eviews загальну модель авторегресії 1 порядку із кубічним трендом:

$$Y=c(1)+c(2)*y(-1)+c(3)*k+c(4)*k^2+c(5)*k^3$$

Будуємо прогноз на проміжок з 10/04/2016 по 5/30/2017 на основі значень даного ряду. Результати демонструємо на рисунку 3.13.

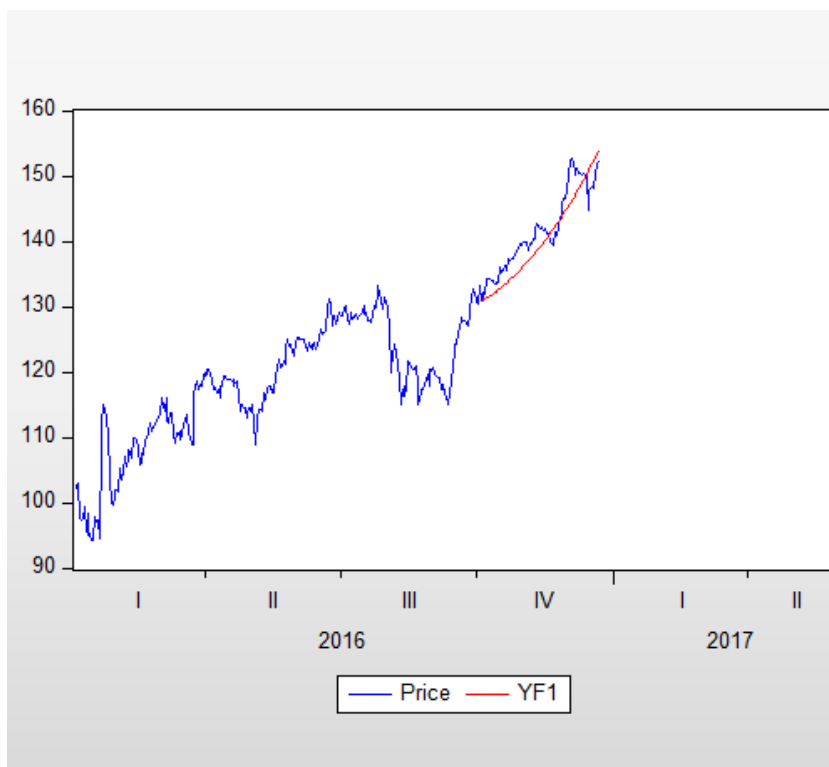


Рисунок 3.13 - Графічне зображення часового ряду «FB» та результату прогнозу на основі моделі  $AR(1)$ +кубічний тренд

СКП = 2.4735; САПП = 1.734864%; Коефіцієнт Тейла = 0.010047;



### 3.2.4 Кубічний тренд+авторегресія 3 порядку

$$\text{res}(k)=c(1)+c(2)*\text{res}(k-1)+c(3)*\text{res}(k-2)+c(4)*\text{res}(k-3)$$

$$C(1) = -0.102968$$

$$C(2) = 0.96137$$

$$C(3) = 0.100911$$

$$C(4) = -0.173895$$

Для даної моделі маємо такі значення критеріїв адекватності:

$$R^2 = 0.833622;$$

$$\sum e^2 = 581.1074;$$

$$DW = 1.948444;$$

$$AIC = 4.077783;$$

Для прогнозу побудуємо в Eviews загальну модель авторегресії 1 порядку із кубічним трендом:

$$Y=c(1)+c(2)*y(-1)+c(3)*y(-2)+c(4)*y(-3)+c(4)*k+c(5)*k^2+c(6)*k^3$$

Будуємо прогноз на проміжок з 10/04/2016 5/30/2017 на основі значень даного ряду. Результати бачимо на рисунку 3.14.

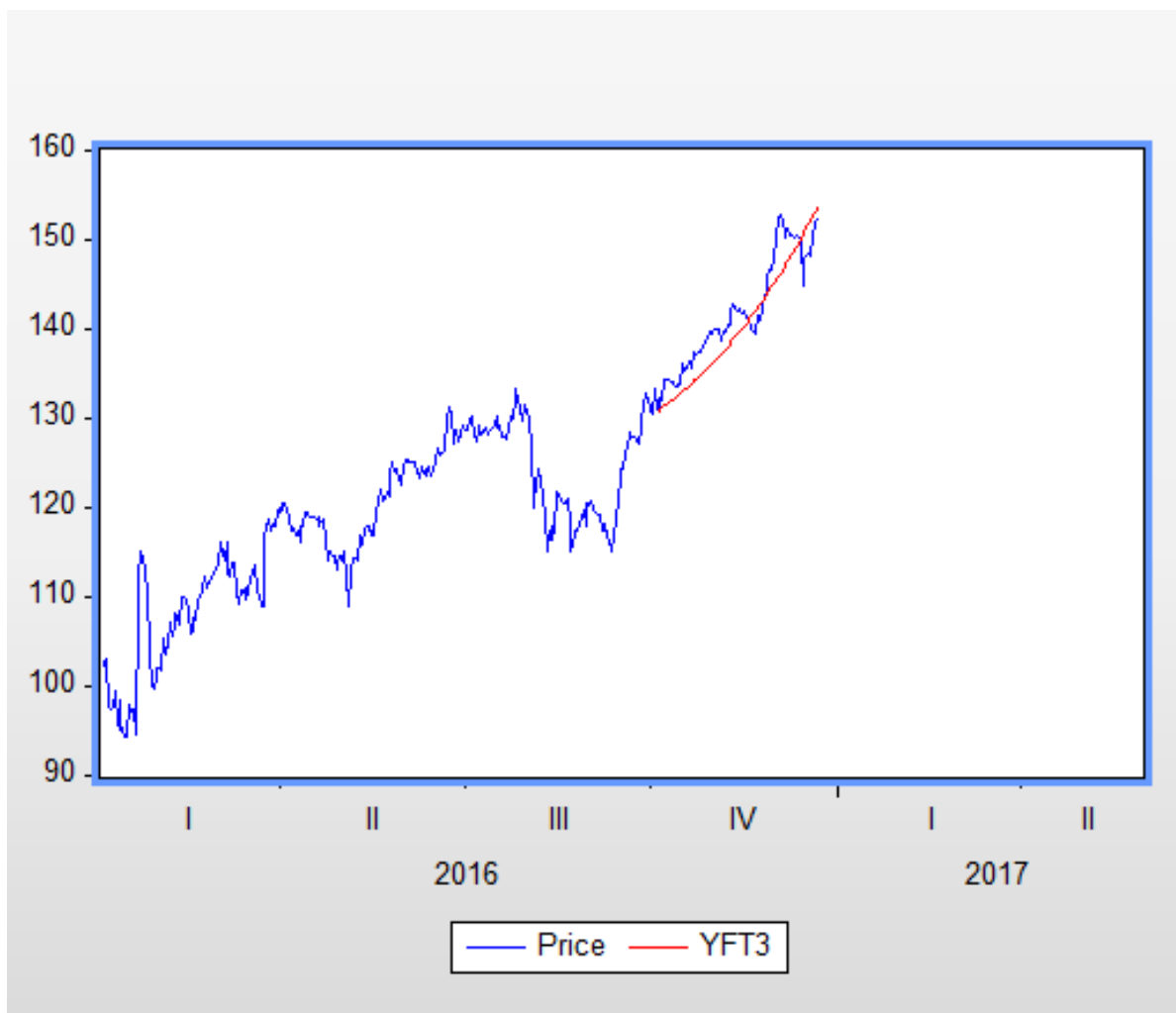


Рисунок 3.14 - Графічне зображення часового ряду «FB» та результату прогнозу на основі моделі  $AR(3)$ +кубічний тренд

СКП = 2.40; САПП = 1.68%; Коефіцієнт Тейла = 0.009794

Таблиця 3.1 - Результати побудови моделей за допомогою Eviews.

Модел ь	Адекватніс ть			Точніст ь прогноз у		
	$R^2$	$\sum e^2$	DW	СК П	САПП	Тейла
Ряд ВАВА 2						
AP(1) + $t^3$	0.92991	454.6827	2.27768	1.21	1.1	0.006577
AP(3) + $t^3$	0.93012 6	437.0856	1.88427 0	1.09	0.99	0.993 22
AP(4) + $t^3$	0.93130	424.2750	1.97505	2.36	2.18584	0.012154
Ряд FB						
AP(1) + $t^3$	0.82990	602.1759	1.69521	2.47	1.73486	2.54514
AP(3) + $t^3$	0.83362	581.1074	1.94844	2.40	1.68	2.47
AP(4) + $t^3$	0.8414	553.5622	1.9746	1.74	1.21157	0.007946

### 3.4 Власне ПЗ

#### 3.4.1 Модель кубічного тренду

$$y = c(1) + c(2)*k + c(3)*k*k + c(4)*k*k*k$$

Будуємо цю ж модель за допомогою власного ПЗ.

Оцінка коефіцієнтів:

$$C(1) = 6298.248$$

$$C(2) = 26.4424$$

$$C(3) = -0.0893$$

$$C(4) = 1.68E-08$$

Продемонструємо роботу ПЗ на рисунку 3.15.

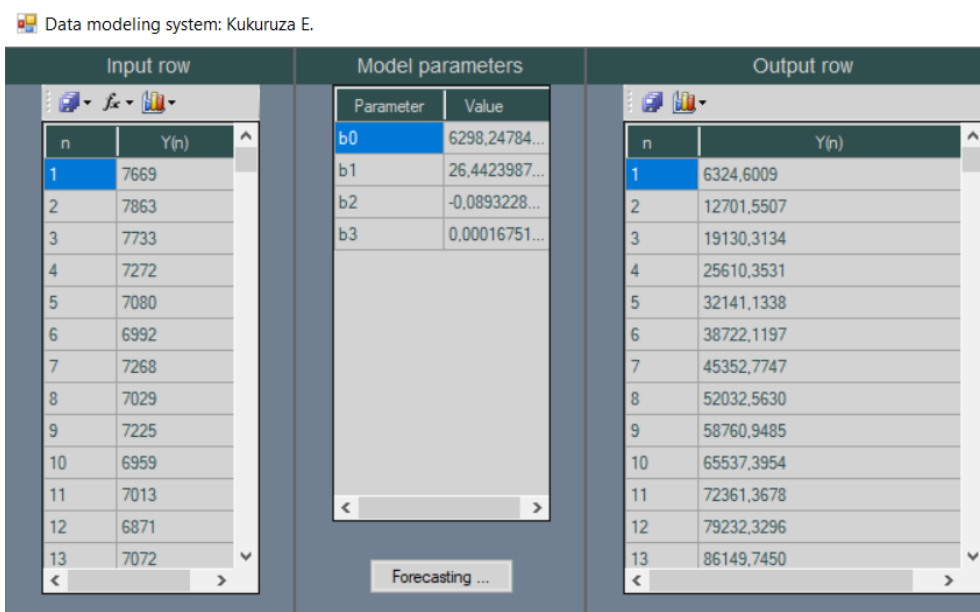


Рисунок 3.15 - Робота власного ПЗ для ряду «ВАВА2»

На рисинку всі дані більші у 100 раз, оскільки програма пристосована для цілих чисел.

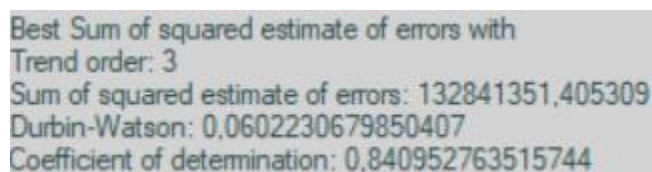
Для даної моделі маємо такі значення критеріїв адекватності:

$$R^2 = 0.840953;$$

$$\sum e^2 = 132841351;$$

$$DW = 0.060223;$$

Вивід даних критеріїв продемонстровано на рисунку 3.16.



Best Sum of squared estimate of errors with  
Trend order: 3  
Sum of squared estimate of errors: 132841351.405309  
Durbin-Watson: 0.0602230679850407  
Coefficient of determination: 0.840952763515744

Рисунок 3.16 - Критерії адекватності моделі кубучного тренду, обраховані за допомогою власного ПЗ

Будуємо прогноз на проміжок з 10/01/2019 по 12/20/2019 (80 днів) на основі значень даного ряду.

Дані результати прогнозу продемонстровано на рисунку 3.17.

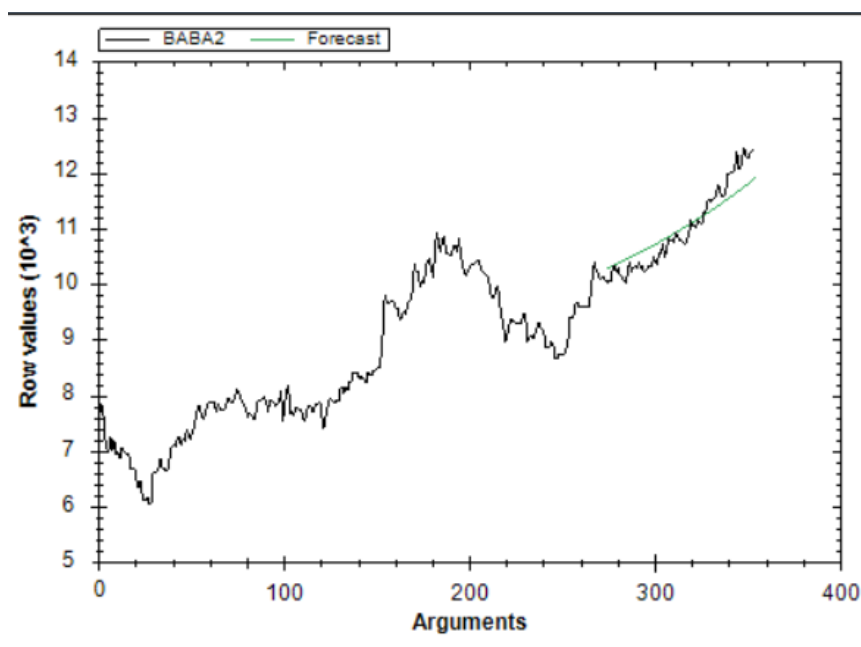


Рисунок 3.17 - Графічне зображення часового ряду «BABA2» та результату прогнозу на основі моделі кубічний тренд

СКП=60.83 СПП=0.2263% САПП=0.2262%

### 3.4.2 Модель АРІКС(4,1,0)

У власному ПЗ ми використовуємо інший метод для прогнозування АРІКС, тому не можемо порівняти коефіцієнти із Eviews.

Для нестационарної частини оцінка коефіцієнтів:

$$C(1) = 16.500176$$

$$C(2) = 0.79508461$$

$$C(3) = -0.0478830$$

$$C(4) = -0.4765793$$

$$C(5) = 0.36867934$$

Для даної моделі маємо такі значення критеріїв адекватності:

$$R^2 = 0.52499;$$

$$\sum e^2 = 10318540;$$

$$DW = 1.8065585;$$

Вивід даних критеріїв продемонстровано на рисунку 3.18.

```
Autoregression: 4
Moving average: 0
Residual order: 3
Sum of squared estimate of errors: 10318540.7950631
Durbin-Watson: 1.80655859549925
Coefficient of determination: 0.524992100557313
Combined statistic: 503.03410073117
```

Рисунок 3.18 - Критерії адекватності, обраховані за допомогою власного ПЗ

Будуємо прогноз на проміжок з 10/01/2019 по 12/20/2019 (80 днів) на основі значень даного ряду.

Дані результати прогнозу продемонстровано на рисунку 3.18.

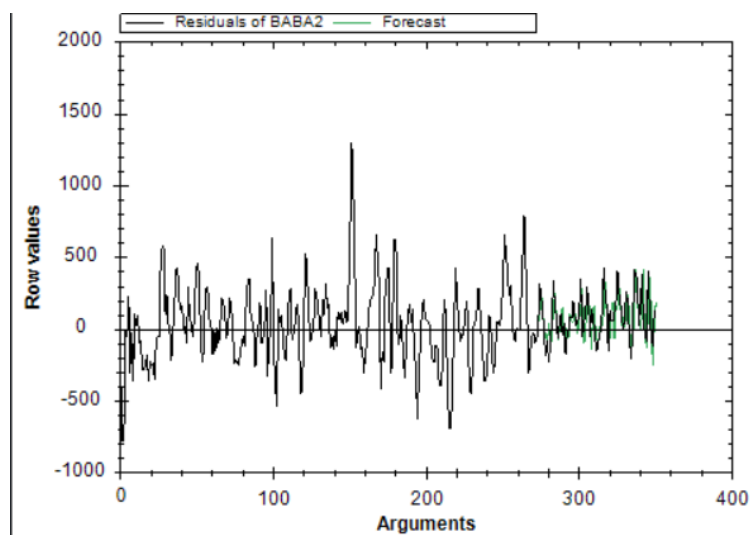


Рисунок 3.19 - Графічне зображення перших різниць ряду «ВАВА2» та результату прогнозу на основі моделі АРІКС(4,1,0)

СКП=146.38045 СПП=0.6599% САПП=0.6599%

### 3.4.3 Модель АРІКС(1,1,0)

У власному ПЗ ми використовуємо інший метод для прогнозування АРІКС, тому не можемо порівняти коефіцієнти із Eviews.

Для нестационарної частини оцінка коефіцієнтів:

$$C(1) = 16.3438103;$$

$$C(2) = 0.62321573;$$

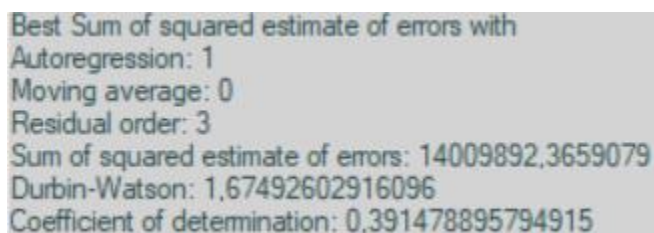
Для даної моделі маємо такі значення критеріїв адекватності:

$$R^2 = 0.3914788;$$

$$\sum e^2 = 14009892;$$

$$DW = 1.6749260;$$

Вивід даних критеріїв продемонстровано на рисунку 3.20.



Best Sum of squared estimate of errors with  
Autoregression: 1  
Moving average: 0  
Residual order: 3  
Sum of squared estimate of errors: 14009892.3659079  
Durbin-Watson: 1.67492602916096  
Coefficient of determination: 0.391478895794915

Рисунок 3.20 - Критерії адекватності, обраховані за допомогою власного ПЗ



Будуємо прогноз на проміжок з 10/01/2019 по 12/20/2019 (80 днів) на основі перших різниць значень даного ряду.

Дані результати прогнозу продемонстровано на рисунку 3.21.

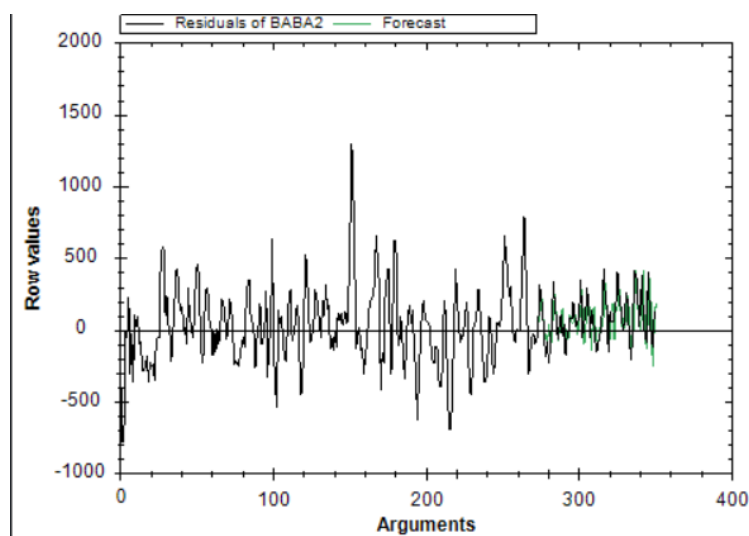


Рисунок 3.21 - Графічне зображення перших різниць ряду «ВАВА2» та результату прогнозу на основі моделі АРІКС(1,1,0)

СКП=163.94954 СПП=0.022098% САПП=0.022098%

#### 3.4.4 Модель АРІКС(3,1,0)

У власному ПЗ ми використовуємо інший метод для прогнозування АРІКС, тому не можемо порівняти коефіцієнти із Eviews.

Для нестационарної частини оцінка коефіцієнтів:

$$C(1) = 26.8995;$$

$$C(2) = 0.71003063;$$

$$C(3) = -0.0947657;$$

$$C(4) = -0.2043984;$$

Для даної моделі маємо такі значення критеріїв адекватності:

$$R^2 = 0.440099865;$$

$$\sum e^2 = 1214494;$$

$$DW = 1.820408;$$

Вивід даних критеріїв продемонстровано на рисунку 3.22.

```
Best Sum of squared estimate of errors with
Autoregression: 3
Moving average: 0
Residual order: 3
Sum of squared estimate of errors: 12144494.218126
Durbin-Watson: 1.82040785612719
Coefficient of determination: 0.440998631478989
```

Рисунок 3.22 - Критерії адекватності, обраховані за допомогою власного ПЗ

Будуємо прогноз на проміжок з 10/01/2019 по 12/20/2019 (80 днів) на основі перших різниць значень даного ряду.

Дані результати прогнозу продемонстровано на рисунку 3.23.

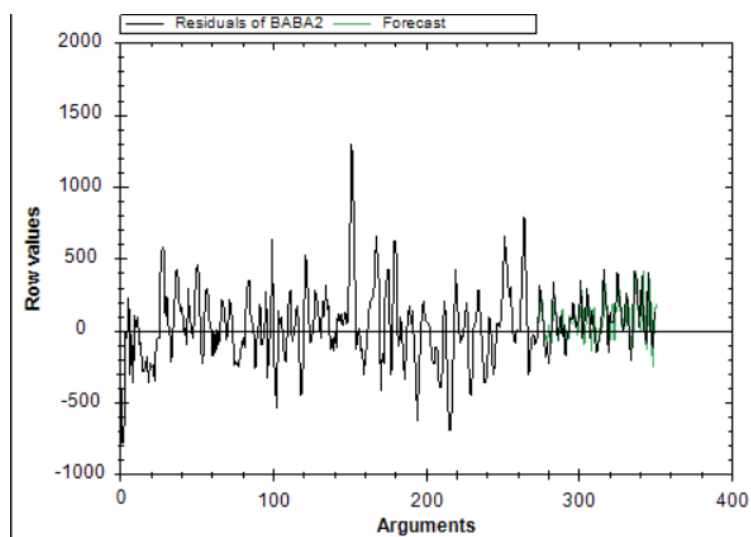


Рисунок 3.23 - Графічне зображення перших різниць ряду «BABA2» та результату прогнозу на основі моделі АРІКС(3,1,0)

СКП=230.74086 СПП=0.42% САПП=0.42%

Для ряду BABA2 моделі кубічний куб порівняємо результати Eviews та власного ПЗ.

Таблиця 3.2 - Результати побудови моделі кубічного тренду в Eviews та за допомогою власного ПЗ.

	Eviews BABA2	ПЗ BABA2
C(1)	62.98248	62.98248
C(2)	0.264424	0.264424
C(3)	-0.000893	-0.000893
C(4)	1.68E-06	1.68E-06
$R^2$	0.840953	0.840953

$\sum e^2$	13284.14	13284.14
DW	0.060223	0.060223
СКП	2.527	2.263
СПП	2.2657	2.263

### 3.4.5 Ряд «Золото»

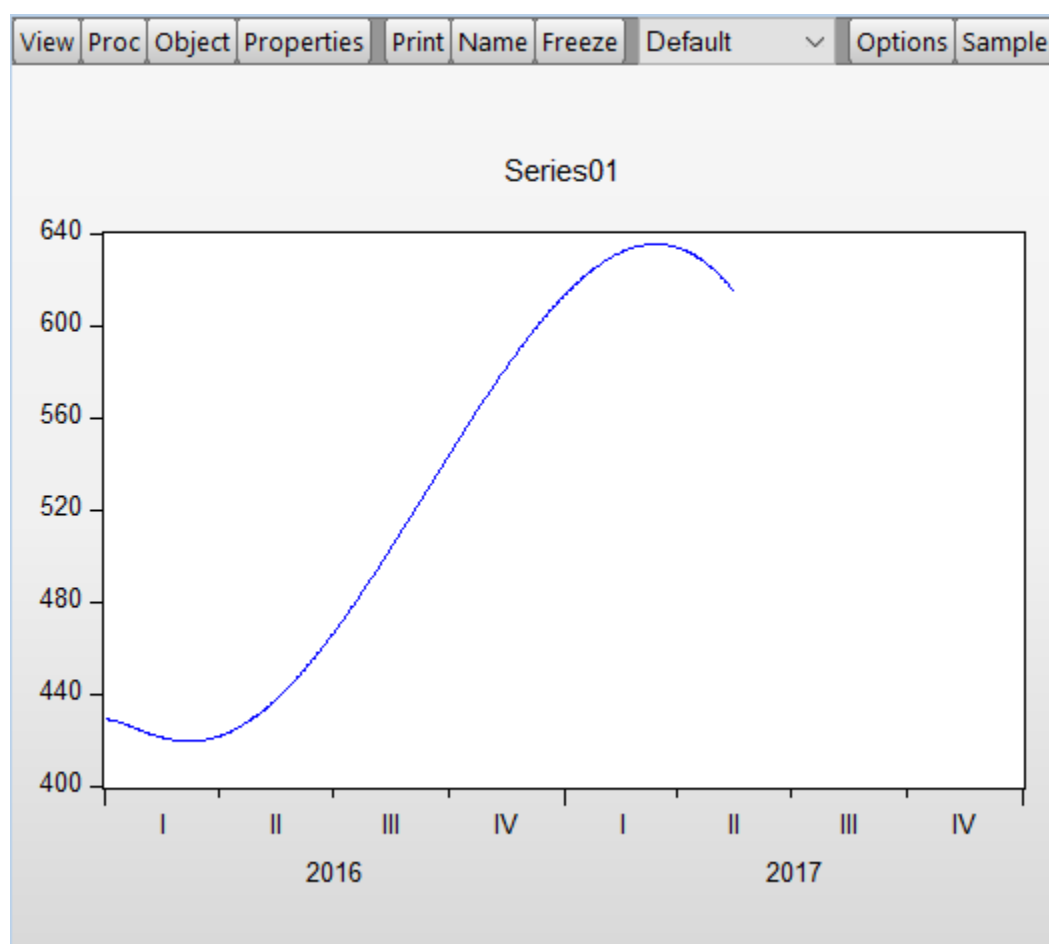


Рисунок 3.24 - Графічне зображення часового ряду «Золото»

Опис характеристик даного ряду:

Mean (середнє) – 525.2309;

Median (медіана) – 523.5177;

Maximum (максимум) – 635.2371;

Skewness (асиметрія) – 0.026083;

Kurtosis (ексцес) – 1.357833;

Jarque-Bera (статистика Жака-Бера) – 56.46316;

Середнє значення не дорівнює нулю.

Для даного ряду теж спробували побудувати моделі

$$y = c(1) + c(2)*k + c(3)*k*k + c(4)*k*k*k$$

i

$$y = c(1) + c(2)*k$$

Після чого для залишків проводимо тест Вайта:

Heteroskedasticity Test: White				
F-statistic	112.3957	Prob. F(6,495)	0.0000	
Obs*R-squared	289.5018	Prob. Chi-Square(6)	0.0000	
Scaled explained SS	1409.485	Prob. Chi-Square(6)	0.0000	
Test Equation:				
Dependent Variable: RESID^2				
Method: Least Squares				
Date: 05/26/20 Time: 01:32				
Sample: 1/01/2016 5/16/2017				
Included observations: 502				
Collinear test regressors dropped from specification				
Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	33.87334	1.319465	25.67204	0.0000
(K)^2	0.017879	0.001248	14.32303	0.0000
(K)*(K^2)	-0.000115	9.30E-06	-12.39618	0.0000
(K)*(K^3)	3.75E-07	3.36E-08	11.14089	0.0000
K	-1.286510	0.072325	-17.78802	0.0000
(K^2)*(K^3)	-5.94E-10	5.80E-11	-10.24363	0.0000
(K^3)^2	3.66E-13	3.83E-14	9.563784	0.0000

### Рисунок 3.25 - Тест вайта для ряду «Золото»

Оскільки ймовірність прийняття нульової гіпотези взагалі дорівнює нулю, тому ми її відхиляємо.

Отже, даний ряд є гетероскедастичним.

#### АРУГ

Для нашого процесу будемо модель авторегресії першого порядку:

$$y(k)=c(1)+c(2)*y(k-1)$$

При використанні теста Уайта визначено, що ряд є гетероскедастичним.

Дослідимо квадрати оцінок залишків.

Побудуємо наступну модель.

$$\hat{\varepsilon}^2(k) = \alpha_0 + \alpha_1 \hat{\varepsilon}^2(k-1) + \alpha_2 \hat{\varepsilon}^2(k-2) + \dots + \alpha_q \hat{\varepsilon}^2(k-q) + v(k),$$

де  $\hat{\varepsilon}^2(k)$  – квадрат похибок на  $k$  місці.

Створила res2 – квадрати залишків. Для них маємо корелелограму.

Для квадратів залишків потрібно будувати AP(1).

$$\text{res2}=c(1) + c(2)*\text{res2}(-1)$$

Основні критерії адекватності мають вигляд:

$$R^2 = 0.234;$$

$$\sum e^2 = 843312.9;$$

$$DW = 1.956;$$

Спрогнозуємо квадрати похибок часового ряду:

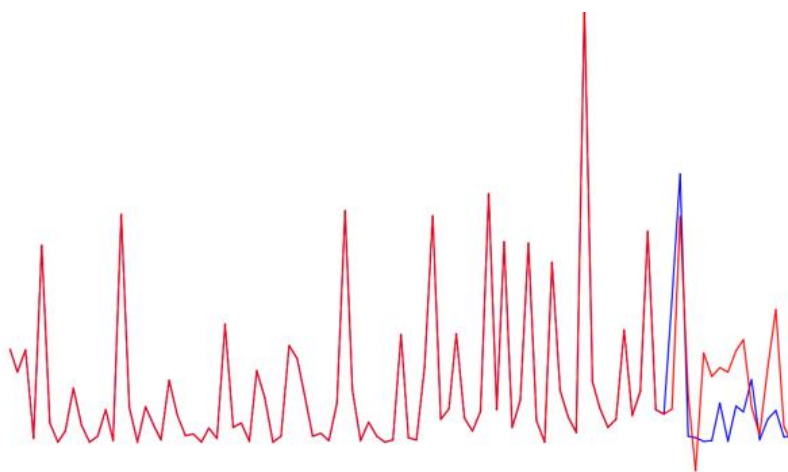


Рисунок 3.26 – Графічне зображення квадратів похибок та прогнозу

$$СКП = 54.34; \text{ САПП} = 23920.48\%; \text{ Коефіцієнт Тейла} = 0.641$$

б) Узагальнена авторегресія з умовною гетероскедастичністю  $УАРУГ(1,1)$ :

$$R^2 = 0.943;$$

$$\sum e^2 = 27948.9;$$

$$DW = 1.746$$

Спрогнозуємо квадрати похибок часового ряду:

$$СКП = 45.8; \text{ САПП} = 12.7\%; \text{ Коефіцієнт Тейла} = 0.023$$

Таблиця 3.3 - Результати побудови моделей за допомогою власного ПЗ.

	$R^2$	$\sum e^2$	DW	СКП	СПП	САПП
AP(1) + $t^3$	0.3914788	14009892	1.6749260	163.94954	0.22098	0.22098
AP(3) + $t^3$	0.440099865	12144494	1.820408	230.74086	0.42	0.42

Продовження таблиці 3.3.

AP(4) + $t^3$	0.52499	10318540	1.8065585	146.38045	0.6599	0.6599
Оцінка умовної дисперсії						
АРУГ(1)	0.234	843312.9	1.956	54.34	23920.48	0.641
УАРУГ(1,1)	0.943	27948.9	1.746	45.8	12.7	0.023

### 3.5 Висновки до розділу

За допомогою Eviews було побудовано різні моделі. В порівняльній таблиці бачимо, що для ряду «ВАВА2» коефіцієнти адекватності моделі найкращих значень набувають при застосуванні моделі кубічного тренду плюс AP(4). Але за моделі кубічного тренду плюс AP(3) будується трішки якісніший прогноз. При застосуванні моделі кубічного тренду плюс AP(1) ми маємо дещо гірші показники, але теж цілком непогані. Для ряду «FB» всі



критерії приймають найкращі значення при застосуванні моделі кубічного тренду+AR(4).

Побудувавши модель кубічного тренду за допомогою власного ПЗ та пакету Eviews, ми отримали однакові оцінки коефіцієнтів та практично однакові значення всіх критеріїв. Отже, програма цілком може скласти конкуренцію пакету Eviews.

Побудувавши моделі за допомогою власного ПЗ для ряду «ВАВА2» теж бачимо, що найадекватнішою є модель кубічного тренду плюс AR(4), але деякі критерії якості прогнозу є трішки кращими у моделі кубічного тренду+AR(3).

Побудувавши для ряду «Золото» за допомогою власного ПЗ моделі АРУГ та УАРУГ, можемо зробити висновок, що критерії набувають дещо кращих значень при моделі УАРУГ.

Отже, можемо зробити висновок, що власна програма досить добре виконує поставлені задачі та цілком готова до подальшого використання. Також бачимо, що побудовані моделі є досить адекватними та роблять гарні прогнози.

## РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

### 4.1 Постановка задачі техніко-економічного аналізу

У роботі застосовується метод ФВА для проведення техніко-економічний аналізу розробки.

Відповідно цьому варто обирати і систему показників якості програмного продукту.

Технічні вимоги до продукту наступні:

- 1) програмний продукт повинен функціонувати на персональних комп'ютерах зі стандартним набором компонент;
- 2) забезпечувати високу швидкість обробки великих об'ємів даних у реальному часі;
- 3) забезпечувати зручність і простоту взаємодії з користувачем або з розробником програмного забезпечення у випадку використання його як модуля;
- 4) передбачати мінімальні витрати на впровадження програмного продукту.

### 4.2 Обґрунтування функцій та параметрів програмного продукту

Функція F0 — розробка програмного продукту, що приймає на вхід дані та виводить рекомендації для прийняття рішення щодо цінних паперів.

Виходячи з цього, можна виділити наступні основні функції ПП:

$F_1$ — вибір мови програмування: мова програмування Python; б) мова програмування C#;

$F_2$ — інтерфейс користувача: а) .NETFramework; б) WindowsForms;

$F_3$ — введення даних: а) введення вручну; б) імпорт даних з файлу.

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 4.1). За допомогою цієї карти побудовано позитивно-негативну матрицю варіантів основних функцій(таблиця 4.1).

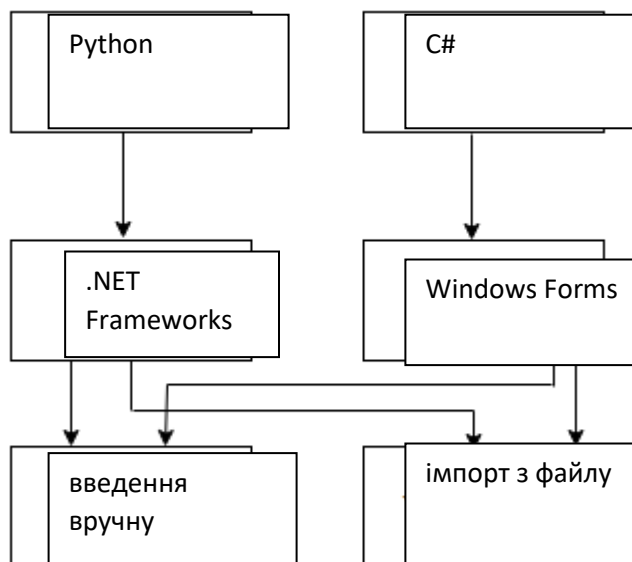


Рис 4.1 – Морфологічна карта

Таблиця 4.1 — Позитивно негативна матриця

Основні функції	Варіанти реалізації	Переваги	Недоліки
$F_1$	$A$	Є всі необхідні бібліотеки	Важкий у використанні

Продовження таблиці 4.1

<i>F1</i>	<i>Б</i>	Простий та зручний у використанні	-
<i>F2</i>	<i>А</i>	Кросплатформеність, полегшений процес завантаження нових даних	Складність при тестуванні
	<i>Б</i>	Нижча вартість та вища швидкість розробки	Більш вузький функціонал.
<i>F3</i>	<i>А</i>	Можна редагувати дані під час введення	Більші затрати часу, можливі помилки під час введення
	<i>Б</i>	Менші затрати часу	Під час вводу немає можливості відредагувати дані

За аналізом позитивно-негативної матриці можна зробити висновок, що при розробці програмного продукту деякі варіанти реалізації функцій потрібно відкинути, бо вони не відповідають поставленій перед програмним продуктом меті.

#### Функція *F1*:

При встановленні необхідних бібліотек і С#, і Python можуть задовільнити наші вимоги, але оскільки С# значно простіший і зручніший у використанні, тому розглядаємо лише цей варіант

#### Функція *F2*:

Вартість програмного продукту залежить від швидкості роботи, тому варіант а) не розглядається.

### Функція F3:

Розглядаємо обидва варіанти, оскільки вони забезпечують вимоги.

Таким чином, будемо розглядати такий варіант реалізації ПП:

1. F1a — F2б — F3a
2. F1a — F2б — F3б.

Функція F3 залежить від двох параметрів X3 і X4(опис даних параметрів наведено в таблиці 4.2).

Для оцінювання функцій обрана система параметрів (табл. 4.2), де визначені гірші, середні і кращі значення параметрів обираємо на основі вимог замовника й умов експлуатації .

Таблиця 4.2. - Основні параметри програмного продукту

Назва Параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Швидкодія мови програмування	X1	нс/Оп	19000	11000	2000
Об'єм пам'яті для коректної роботи	X2	Мб	32	16	8
Час навчання	X3	мс	2000	420	60
Потенційний об'єм програмного коду	X4	кількість рядків коду	2000	1500	1000

Графічні характеристики параметрів

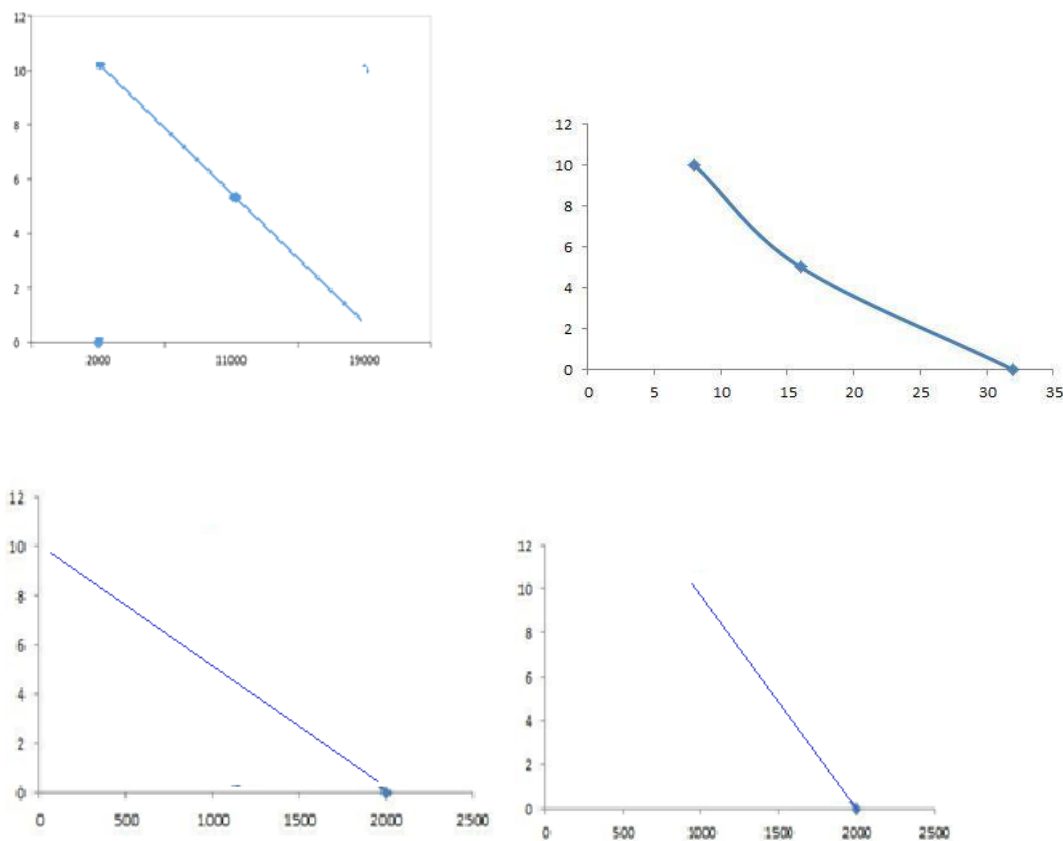


Рис 4.2 Графіки зміни параметрів X1, X2, X3, X4.

Найбільший ранг –1, найменший –4.

Таблиця 4.3 — Результати ранжування параметрів

Позначення параметра	Одиниці виміру	Ранг параметра за оцінкою експерта	Сума рангів $R_i$	Відхилення $\Delta_i$	$\Delta^2_i$



Продовження таблиці 4.4

X2 і X3	>	>	<	>	>	>	>	>	1.5
X2 і X4	>	>	>	>	>	>	>	>	1.5
X3 і X4	<	<	>	<	>	>	<	<	0.5

Таблиця 4.5 - Розрахунок вагомості параметрів

Параметри $x_i$	Параметри $x_j$				Перша ітерація		Друга ітерація		Третя ітерація	
	1	2	3	4	$b_i$	$K_{bi}$	$b_i^1$	$K_{bi}$	$b_i^2$	$K_{bi}$
X1	1.0	0.5	1.5	1.5	4.5	0.280	16.0	0.272	58.875	0.274
X2	1.5	1.0	1.5	1.5	5.5	0.344	21.25	0.360	77.5	0.36
X3	0.5	0.5	1.0	0.5	2.5	0.156	9.25	0.157	34	0.158
X4	0.5	0.5	1.5	1.0	3.5	0.22	12.25	0.209	44.75	0.208
Всього:					16.0	1	58.75	1	215.125	1

Таблиця 4.6 – Розрахунок показників рівня якості варіантів реалізації  
основних функцій ПП –

Основні функції	Варіант реалізації	Параметри –	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	А	X1	2000	10	0.274	2.74
F2	Б	X2	8	10	0.36	3.6
F3	А	X3	1700	2	0.158	0.316
		X4	1400	2	0.208	0.416



Продовження таблиці 4.6.

	Б	Х3	60	10	0.158	1.58
		Х4	1000	5	0.208	1.04

$$K_{K1} = 2.74 + 3.6 + 0.316 + 0.416 = 7.072$$

$$K_{K2} = 2.74 + 3.6 + 1.58 + 1.04 = 8.96$$

Другий варіант виявився кращим, оскільки коефіцієнт технічного рівня має найбільше значення.

#### 4.3. Економічний аналіз варіантів розробки програмного продукту

Усі варіанти включають в себе наступні завдання:

1. Розробка методів підготовки даних для їх використання в ПП:

Для першого варіанту реалізації використовуємо нормалізацію;

Для другого варіанту – цифрову фільтрацію

2. Розробка проекту програмного продукту;

3. Розробка програмної оболонки;

Для першого завдання ((при реалізації варіанту 1А алгоритм групи складності 3, ступінь новизни В, вид використаної інформації — НДІ)  $T_p = 15$ ,  $K_{\Pi} = 1.26$ ,  $K_{СК} = 1$ ,  $K_{СТ.М} = 1$

$$T_1^1 = 15 * 1.26 = 18.9 \text{ людино-днів}$$

Для першого завдання (при реалізації варіанту 1Б, (алгоритм групи складності 1, ступінь новизни А, вид використаної інформації — НДІ)) $T_p = 25$ ,

$K_{\Pi} = 1.26$  для НДІ,  $K_{СК} = 1$ ,  $K_{СТ.М} = 1.6$ .

$$T_1^2 = 25 * 1.26 * 1.6 = 50.4 \text{ людино-днів}$$

Для другого завдання (ступінь новизни А, група складності алгоритму 1)  $T_p = 80$  людино-днів,  $K_{\Pi} = 1.6$ ,  $K_{СК} = 1$ ,  $K_{СТ} = 0.8$ .

$$T_2 = 80 * 1.6 * 0.8 = 102.4 \text{ людино-днів.}$$

Для третього завдання (ступінь новизни Б, група складності алгоритму 3)  $T_p = 26$  людино-днів,  $K_{\Pi} = 0.7$ ,  $K_{СТ} = 0.8$ :

$$T_3 = 26 * 0.7 * 0.8 = 14.56 \text{ людино-днів}$$

Трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми

$$T_I = (18.9 + 102.4 + 14.56) \cdot 8 = 1086.88 \text{ людино-годин;}$$

$$T_{II} = (50.4 + 102.4 + 14.56) \cdot 8 = 1338.88 \text{ людино-годин;}$$

Найбільш високу трудомісткість має варіант II.

В розробці бере участь один фінансовий інженер з окладом 31000 грн.

Зарплата за годину:

$$C = \frac{31000}{1 * 21 * 8} = 184.52 \text{ грн.}$$

Зарплата розробників

$$I. C_{3II} = 184.52 * 1086.88 = 200551 \text{ грн.}$$

$$\text{II. } C_{3П} = 184.52 * 1338.88 = 247050 \text{ грн.}$$

Відрахування на соціальний внесок становить 22,0%:

$$\text{I. } C_{ВІД} = 200551 * 0.22 = 44121 \text{ грн.}$$

$$\text{II. } C_{ВІД} = 247050 * 0.22 = 54351 \text{ грн.}$$

Обчислимо витрати на оплату однієї машино-години. Оскільки ЕОМ обслуговує один інженера з окладом 31000 грн., з коефіцієнтом зайнятості 0.2, то для однієї машини маємо:

$$C_{Г} = 12 * M * K_3 = 12 * 31000 * 0.2 = 74400 \text{ грн.}$$

З урахуванням додаткової заробітньої плати:

$$C_{3П} = C_{Г} * (1 + K_3) = 74400 * (1 + 0.2) = 89280 \text{ грн.}$$

Відрахування на соціальний внесок:

$$C_{ВІД} = C_{3П} * 0.22 = 89280 * 0.22 = 19641.6 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 160000 грн.

$$C_A = K_{ТМ} * K_A * Ц_{ПР} = 1.15 * 0.25 * 160000 = 46000 \text{ грн.,}$$

Витрати на ремонт та профілактику:

$$C_P = K_{ТМ} * Ц_{ПР} * K_P = 1.15 * 160000 * 0.05 = 9200 \text{ грн.}$$

Ефективний годинний фонд часу ПК:

$$T_{ЕФ} = (D_K - D_B - D_C - D_P) * t_3 * K_B = (365 - 112 - 8 - 16) * 8 * 0.9 = 1648.8 \text{ годин}$$

Витрати на оплату електроенергії (з урахуванням ПдВ):

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} * N_{\text{С}} * K_3 * C_{\text{ЕН}} = 1648.8 * 0.2 * 2 * 1.75 = 1154.16 \text{ грн.},$$

Накладні витрати

$$C_{\text{Н}} = C_{\text{ПР}} * 0.67 = 160000 * 0.67 = 107200 \text{ грн.}$$

Річні експлуатаційні втрати:

$$C_{\text{ЕКС}} = 89280 + 19641.6 + 46000 + 9200 + 1154.16 + 107200 = 272475.76 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнює:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 272475.76 / 1648.8 = 165.26 \text{ грн./год.}$$

Витрати на оплату машинного часу:

$$\text{I. } C_{\text{М}} = 165.26 * 1086.88 = 179614.54 \text{ грн.}$$

$$\text{II. } C_{\text{М}} = 165.26 * 1338.88 = 221263.309 \text{ грн.}$$

Накладні витрати складають 67% від заробітної плати:

$$\text{I. } C_{\text{Н}} = 179614.54 * 0.67 = 120341.742 \text{ грн.}$$

$$\text{II. } C_{\text{Н}} = 221263.309 * 0.67 = 148246.417 \text{ грн.}$$

Вартість розробки програмного продукту:

$$\text{I. } C_{\text{ПП}} = 200551 + 44121 + 179856.9 + 120341.742 = 544870.642 \text{ грн.}$$

$$\text{II. } C_{\text{ПП}} = 247050 + 54351 + 221557.862 + 148246.417 = 671205.279 \text{ грн.}$$

Розрахуємо коефіцієнт техніко-економічного рівня:

$$K_{\text{ТЕР1}} = 7.072 / 544870.642 = 1.298 * 10^{-5};$$

$$K_{\text{ТЕР2}} = 8.96 / 671205.279 = 1.34 * 10^{-5};$$

#### 4.4 Висновки до розділу

Звідси можна робити висновок, що найбільш ефективним є другий варіант з коефіцієнтом техніко-економічного рівня  $1.34 * 10^{-5}$ .

З альтернатив, що залишилися після відбору, було отримано два варіанти. Доцільним є використання другого варіанту реалізації продукту. Цьому варіанту відповідає мова C#, WindowsForms і импорт даних з файла. Таким чином ми отримали універсальний засіб створення системи підтримки прийняття рішень для аналізу ринкових фінансових ризиків.

## ВИСНОВКИ

В роботі проаналізовано методи, за допомогою яких можна моделювати фінансові нелінійні нестационарні процеси та які забезпечують можливості оцінювання короткострокових прогнозів економічних та інших показників. На їх основі реалізовано програмний продукт, за допомогою якого можливо визначати найкращі моделі для відповідних типів даних з мінімальними витратами ресурсів. У результаті було побудовано низку різнотипових регресійних моделей. На основі цього оцінок параметрів методів було здійснено порівняльний аналіз методів між собою. Дані методи та моделі було застосовано для трьох рядів: BABA2, FB та GOLD. Критеріями якості побудованих моделей були коефіцієнт детермінації, сума квадратів похибок та статистика Дарбіна-Уотсона. За критерії якості прогнозування були взяті: середньо-квадратична похибка, абсолютна середня абсолютна похибка у процентах та коефіцієнт Тейла. Отримані результати засвідчили різну якість побудованих моделей та можливість їхнього практичного застосування в рамках розробленого продукту. Водночас варто зазначити, що для кожної конкретної ситуації неможливо заздалегідь передбачити, який з методів дасть найкращий прогноз. Отже, загальний підхід до прогнозування нелінійних нестационарних фінансово-економічних процесів має бути комплексним та базуватися на сучасних методах системного аналізу та застосуванні ідеологічно різних математичних моделей. При цьому побудований прогноз стає підґрунтям для прийняття обґрунтованих управлінських рішень.

## ЛІТЕРАТУРА


1. Бідюк П.І., Коршевніук Л.О. Проектування комп'ютерних інформаційних систем підтримки прийняття рішень: навч. посіб. / ННК «Інститут прикладного системного аналізу» Національний технічний університет України «Київський політехнічний інститут», 2010. 340 с.
2. Пугачов О.М. Аналіз деяких моделей для прогнозування нестационарних процесів. Київ: Інтернаука, 2016. 197-200 с.
3. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge, MA: MIT Press, 2017. 775 p.
4. Lecun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation*. 1989. Vol. 1, No. 4. 541-551.
5. Лобанов А. А., Чугунов В. А. Энциклопедия финансового риск-менеджмента. Москва: Альпина Паблишер, 2003. 786 с.
6. Бідюк П.І. Часові ряди: моделювання та прогнозування / Київ: ЕКМО, 2004. 144 с.
7. Дрейпер Н., Смит Г. Финансы и статистика: в 2-х кн. Москва: 1986. Т.2: Прикладной регрессионный анализ. 366 с.

8. Бідюк П. І. Часові ряди: моделювання і прогнозування: монографія / Київ: ЕКМО, 2003. 144 с.
9. Trigg D.W., Leach A.G. Exponential smoothing with an adaptive response rate: journal article / Operational Research Society, 1967.
10. Holt C.C. Forecasting trends and seasonals by exponentially weighted moving averages / O.N.R. Memorandum, Carnegie Inst. of Technology. 1957.
11. Brown R.G., Meyer R.F. The fundamental theorem of exponential smoothing /Oper. Res. 1961.
12. Brown R.G. Smoothing forecasting and prediction of discrete time series. / N.Y., 1963.
13. Канторович Г.Г. Анализ временных рядов: учебник / Москва: 2002. 544с.
14. Магнус Я. Р. Эконометрика. Начальный курс : учебник / Москва : Дело, 2004. 576 с.
15. Бідюк П. І., Терентьев О. М., Просянкін-Жарова Т. І. Прикладна статистика: підручник / Вінниця: ПП «ТД» Едельвейс і К», 2013. 304с.
16. Пугачов О.М. Аналіз деяких моделей для прогнозування нестационарних процесів. Київ: Інтернаука, 2016. 197-200 с.
17. Бідюк П. І. Система підтримки прийняття рішень для аналізу і прогнозування стану підприємства: навч посіб./ Київ — 2013. С. 128–136.
18. Бідюк П.І., Коршевнюк Л.О. Проектування комп'ютерних інформаційних систем підтримки прийняття рішень: Навчальний посібник/ Київ: ННК “ІПСА” НТУУ “КПІ”, 2010. 138 -205 с.
19. Engle F.R. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation, 1982. pp. 987-1007.



- 20.** Єріна А. М. Статистичне моделювання та прогнозування: Навч. посібник / Київ: КНЕУ, 2001. 170 с.

## ДОДАТОК А




# Моделі і методи прогнозування нелінійних нестационарних процесів в економіці та фінансах

Виконала студентка групи КА-63

Кукуруза Єлизавета

Науковий керівник: д.т.н., професор кафедри ММСА Бідюк Петро Іванович



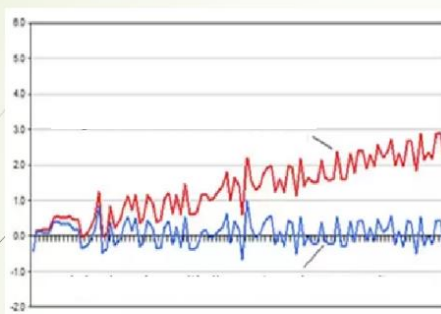
Об'єкт дослідження: Нелінійні нестационарні процеси в економіці та фінансах.

Предмет дослідження: Методи і моделі опису процесів в економіці.

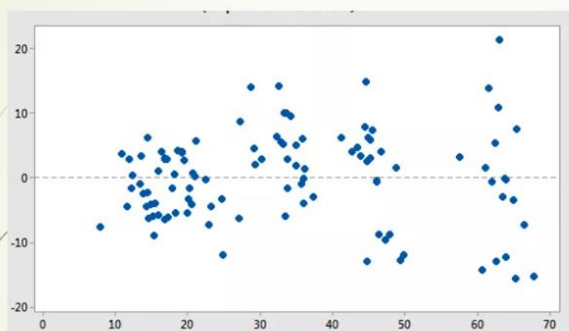
Мета: порівняти методи і моделі прогнозування на основі економічних даних.

Постановка задачі:

- Дослідити ряди на інтегрованність/гетероскедастичність
- Побудувати по декілька модель для кожного ряду за допомогою Eviews
- Розробити ПЗ для прогнозування процесів
- Побудувати декілька моделей за допомогою власного ПЗ
- Порівняти отримані результати на основі критеріїв адекватності моделей та якості прогнозування



Приклад ряду із трендом та цього ж ряду, позбавленого тренду



### Приклад гетероскедастичного процесу

Критерії адекватності математичних моделей:

- **t-Statistic** (t-статистика Стюдента)

$$t_a = \frac{\hat{a} - a_0}{SE_{\hat{a}}},$$

- **R-squared**  $R^2$  (коефіцієнт детермінації)

$$R^2 = \frac{var(\hat{y})}{var(y)}$$

- **Sum square resid** (Сума квадратів похибок моделі) SSE

$$SSE = \sum_{t=1}^N (y - \hat{y})^2$$

- **Akaike info criterion** (Інформаційний критерій Акайке (AIC))

$$AIC = -\frac{2 \cdot l}{T} + \frac{2 \cdot k}{T}$$

- **Durbin-Watson stat** (Коефіцієнт Дурбіна-Уотсона (DW))

$$DW = \frac{\sum_{t=2}^N (\varepsilon_t - \varepsilon_{t-1})^2}{\sum_{t=1}^N \varepsilon_t^2}$$

- Коефіцієнт Тейла

$$U = \frac{\sqrt{\frac{1}{N} \sum_{k=1}^N [y(k) - \hat{y}(k)]^2}}{\sqrt{\frac{1}{N} \sum_{k=1}^N y^2(k) + \frac{1}{N} \sum_{k=1}^N \hat{y}^2(k)}}$$

Критерії якості прогнозу:

- S.E. of regression середньоквадратична похибка (СКП)

$$СКП = \sqrt{E((y - \hat{y})^2)} = \sqrt{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}}$$

- Сума похибок (СП)

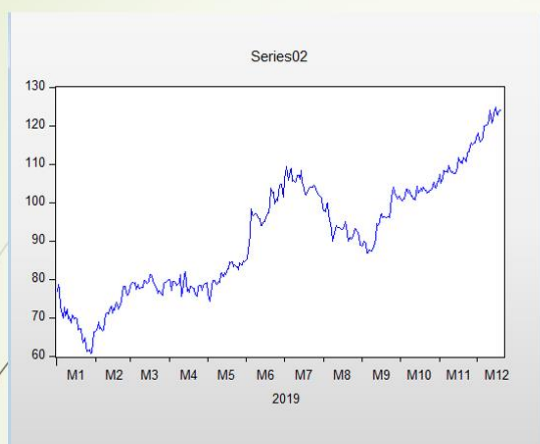
$$СП = \frac{1}{N} \sum_{k=1}^N [y(k) - \hat{y}(k)] = \frac{1}{N} \sum_{k=1}^N e(k)$$

- Середня похибка у процентах (СПП)

$$СПП = \frac{1}{N} \sum_{k=1}^N \frac{[y(k) - \hat{y}(k)]}{y(k)} \times 100\%,$$

- Середнє абсолютне значення похибки в процентах (САПП)

$$САПП = \frac{1}{N} \sum_{k=1}^N \frac{|y(k) - \hat{y}(k)|}{|y(k)|} \times 100\%$$



Модель  $t^3$ :

$$y = c(1) + c(2)*k + c(3)*k*k + c(4)*k*k*k$$

Модель  $AR(1) + t^3$ :

$$Y = c(1) + c(2)*y(-1) + c(3)*k + c(4)*k^2 + c(5)*k^3$$

Модель  $AR(3) + t^3$ :

$$Y = c(1) + c(2)*y(-1) + c(3)*y(-2) + c(4)*y(-3) + c(5)*k + c(6)*k^2 + c(7)*k^3$$

Модель  $AR(4) + t^3$ :

$$Y = c(1) + c(2)*y(-1) + c(3)*y(-2) + c(4)*y(-3) + c(5)*y(-4) + c(6)*k + c(7)*k^2 + c(8)*k^3$$

Прогнозування на основі різних моделей:



Модель  $t^3$



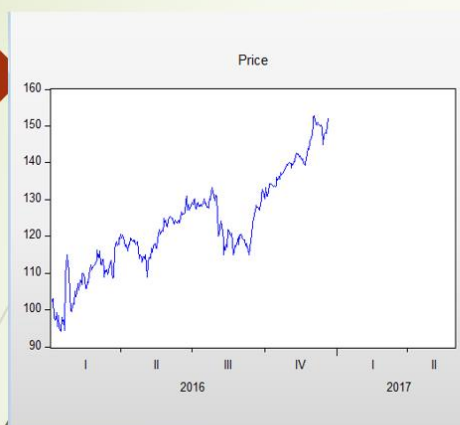
Модель  $AR(3) + t^3$ :



Модель  $AR(1) + t^3$



Модель  $AR(4) + t^3$ :



Модель  $t^3$ :

$$y = c(1) + c(2)*k + c(3)*k*k + c(4)*k*k*k$$

Модель  $AR(1) + t^3$ :

$$Y = c(1) + c(2)*y(-1) + c(3)*k + c(4)*k^2 + c(5)*k^3$$

Модель  $AR(3) + t^3$ :

$$Y = c(1) + c(2)*y(-1) + c(3)*y(-2) + c(4)*y(-3) + c(4)*k + c(5)*k^2 + c(6)*k^3$$

Модель  $AR(4) + t^3$ :

$$Y = c(1) + c(2)*y(-1) + c(3)*y(-2) + c(4)*y(-3) + c(5)*y(-4) + c(6)*k + c(6)*k^2 + c(7)*k^3$$

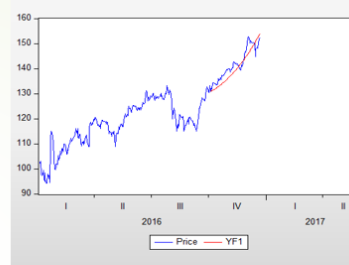
Прогнозування на основі різних моделей:



Модель  $t^3$



Модель  $AR(3) + t^3$ :



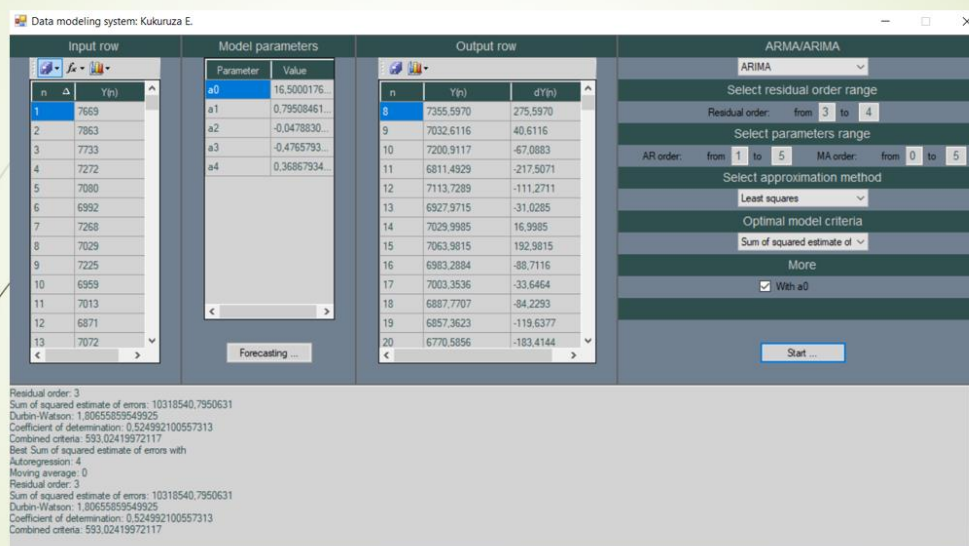
Модель  $AR(1) + t^3$



Модель  $AR(4) + t^3$ :

Ряд ВАВА 2	$R^2$	$\sum e^2$	DW	СКП	САПП	Тейла
AR(1) + $t^3$	0.9299 1	454.68 27	2.2776 8	1.21	1.1	0.006577
AR(3) + $t^3$	0.9301 26	437.08 56	1.8842 70	1.09	0.99	0.9932 2
AR(4) + $t^3$	0.9313 0	424.27 50	1.9750 5	2.36	2.1858 4	0.012154
Ряд FB						
AR(1) + $t^3$	0.8299 0	602.17 59	1.6952 1	2.47	1.7348 6	2.5451 4
AR(3) + $t^3$	0.8336 2	581.10 74	1.9484 4	2.40	1.68	2.47
AR(4) + $t^3$	0.8414	553.56 22	1.9746	1.74	1.2115 7	0.007946

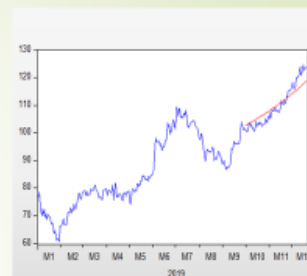
Приклад роботи власного ПЗ:



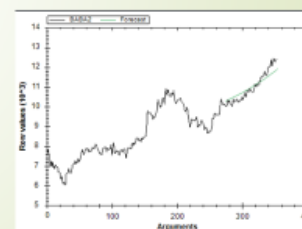
Для моделі  $t^3$  співставлення оцінених коефіцієнтів та отриманих критеріїв:

	Eviews	ПЗ
	BA(2)	BA(2)
C(1)	62.98248	62.98248
C(2)	0.264424	0.264424
C(3)	-0.000893	-0.000893
C(4)	1.68E-06	1.68E-06
$R^2$	0.840953	0.840953
$\sum e^2$	13284.14	13284.14
DW	0.060223	0.060223
СКП	2.527	2.263
СПП	2.2657	2.263

За допомогою Eviews

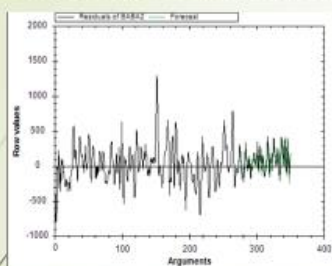


За допомогою власного ПЗ

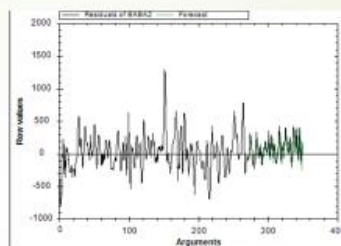




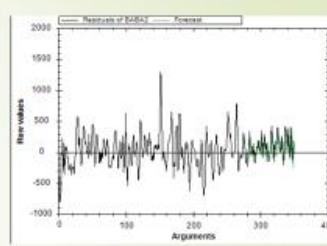
## Графічні зображення перших різниць ряду та прогнозів на основі різних моделей



на основі моделі АРІКС(1,1,0)

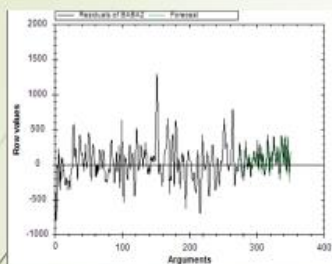


на основі моделі АРІКС(3,1,0)

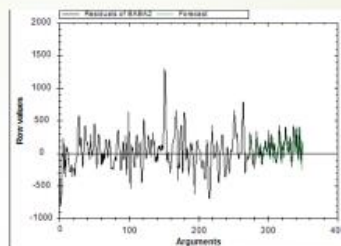


на основі моделі АРІКС(4,1,0)

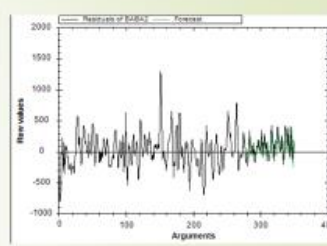
## Графічні зображення перших різниць ряду та прогнозів на основі різних моделей



на основі моделі АРІКС(1,1,0)



на основі моделі АРІКС(3,1,0)



на основі моделі АРІКС(4,1,0)

Порівняння різних моделей,  
побудованих та оцінених за допомогою  
власного ПЗ:

	$R^2$	$\sum e^2$	DW	СКП	СПП	САПП
AP(1) +t <sup>3</sup>	0.39147 88	1400989 2	1.67492 60	163.949 54	0.22098	0.22098
AP(3) +t <sup>3</sup>	0.44009 9865	1214449 4	1.82040 8	230.740 86	0.42	0.42
AP(4) +t <sup>3</sup>	0.52499	1031854 0	1.80655 85	146.380 45	0.6599	0.6599

#### Висновки:

- Проведено огляд методів і моделей прогнозування нелінійних нестационарних процесів.
- Оцінено дані моделі за допомогою програми Eviews.
- Розроблено власне ПЗ та використано.
- Застосовано ПЗ для прогнозування на основі різних моделей.
- Співставлено й проаналізовано результати, отримані на основі різних моделей.



Дякую за увагу !

## ДОДАТОК Б

```

using System;

namespace ElizabethDataModeling.Algorithms
{
    public enum Methods { LSM = 0, RLSM = 1 };
    public class Auxiliary
    {
        public static Matrix MatrixForNonstatTest(double[] dataAr, bool IsAVG, bool
IsTrend, string ParamsAR)
        {
            int CountColumns = 1;
            if (IsAVG) CountColumns++;
            if (IsTrend) CountColumns++;

            int CountRows;
            int startDataAr;
            string[] prmAR;
            double[] diffDataAr;
            if (!String.IsNullOrEmpty(ParamsAR))
            {
                prmAR = ParamsAR.Split(';');
                CountColumns = CountColumns + prmAR.Length;
                CountRows = dataAr.Length - 1 - int.Parse(prmAR[prmAR.Length - 1]);
                startDataAr = 1 + int.Parse(prmAR[prmAR.Length - 1]);
                double[] tmp = new double[dataAr.Length];
                dataAr.CopyTo(tmp, 0);
                Convertation.DifferRow(ref tmp, 1);
                diffDataAr = new double[tmp.Length - 1];
                Array.Copy(tmp, 1, diffDataAr, 0, diffDataAr.Length);
            }
            else
            {
                prmAR = new string[0];
                diffDataAr = new double[0];
                CountRows = dataAr.Length - 1;
                startDataAr = 1;
            }

            Matrix H = new Matrix(CountRows, CountColumns);
            int nowClmn = 0;
            if (IsAVG)
            {
                for (int i = 0; i < H.CountRows; i++) { H[i, 0] = 1; }
                nowClmn++;
            }
            if (IsTrend)
            {
                for (int i = 0; i < H.CountRows; i++)
                {
                    H[i, nowClmn] = startDataAr + i + 1;
                }
            }
        }
    }
}

```

```

        nowClmn++;
    }
    for (int i = 0; i < H.CountRows; i++)
    {
        H[i, nowClmn] = dataAr[startDataAr + i - 1];
    }
    nowClmn++;
    if (!String.IsNullOrEmpty(ParamsAR))
    {
        for (int k = 0; k < prmAR.Length; k++)
        {
            int p = int.Parse(prmAR[k]);
            for (int i = 0; i < H.CountRows; i++)
            {
                H[i, nowClmn + k] = diffDataAr[i + p];
            }
        }
    }

    return H;
}

public static double NormVec(Matrix Mtrx)
{
    double res = 0;
    for (int i = 0; i < Mtrx.CountRows; i++)
    {
        res += Mtrx[i, 0] * Mtrx[i, 0];
    }
    return Math.Sqrt(res);
}

static public Matrix LSM(Matrix H, Matrix Y)
{
    Matrix temp = Matrix.Transpose(H);
    temp = Matrix.Increase(temp, H);
    temp = Matrix.Inverse(temp);
    temp = Matrix.Increase(temp, Matrix.Transpose(H));
    Matrix B = Matrix.Increase(temp, Y);
    return B;
}

static public Matrix RLSM(Matrix H, Matrix Y, double beta)
{
    Matrix B = new Matrix(H.CountColumns, 1);
    Matrix P = Matrix.DiagonalMatrix(H.CountColumns, beta);
    Matrix Row;
    for (int i = 0; i < H.CountRows; i++)
    {
        Row = new Matrix(H.RowArray(i));
        Row = Matrix.Transpose(Row);
        P = PFunc(P, Row);
        B = BFunc(P, Row, B, Y[i, 0]);
    }
    return B;
}

static private Matrix PFunc(Matrix P, Matrix RowH)

```

```

    {
        Matrix chs = Matrix.Transpose(RowH);
        chs = Matrix.Increase(P, chs);
        chs = Matrix.Increase(chs, RowH);
        chs = Matrix.Increase(chs, P);

        Matrix znm = Matrix.Transpose(RowH);
        znm = Matrix.Increase(P, znm);
        znm = Matrix.Increase(RowH, znm);
        znm[0, 0] = znm[0, 0] + 1;

        Matrix Res = Matrix.IncrConst(1 / znm[0, 0], chs);
        Res = Matrix.Diff(P, Res);
        return Res;
    }

    static private Matrix BFunc(Matrix P, Matrix RowH, Matrix B, double dimY)
    {
        Matrix temp = Matrix.Increase(RowH, B);
        temp[0, 0] = dimY - temp[0, 0];
        temp = Matrix.Increase(temp, RowH);
        temp = Matrix.Transpose(temp);
        temp = Matrix.Increase(P, temp);
        B = Matrix.Add(B, temp);
        return B;
    }
}

namespace ElizabethDataModeling.Algorithms
{
    public static class Constants
    {
        public const double EPS = 0.0001;

        public static int Power(int Number, int Exponent)
        {
            var result = 1;
            for (var i = 0; i < Exponent; i++)
                result *= Number;
            return result;
        }
    }
}

using System;

namespace ElizabethDataModeling.Algorithms
{
    public class Convertation
    {
        public static bool NormalizationRow(ref double[] dataAr, int typeNorm)
        {
            if (IsUndefinedElement(dataAr)) return false;
            if (typeNorm != -1 & typeNorm != 0) return false;
            double[] MaxEl = MaxElementAr(dataAr);
            double[] MinEl = MinElementAr(dataAr);
            double znm = MinEl[0] - MaxEl[0];

```

```

        if (znam == 0) return false;
        for (int i = 0; i < dataAr.Length; i++)
        {
            if (typeNorm == 0) dataAr[i] = (MinEl[0] - dataAr[i]) / znam;
            else dataAr[i] = 2 * (MinEl[0] - dataAr[i]) / znam - 1;
        }
        return true;
    }
}
public static double[] MaxElementAr(double[] mass)
{
    double[] outMass = { mass[0], 0 };
    if (IsUndefinedElement(mass))
    {
        outMass[0] = double.NaN;
        outMass[1] = -1;
        return outMass;
    }
    for (int i = 1; i < mass.Length; i++)
    {
        if (outMass[0] < mass[i])
        {
            outMass[0] = mass[i];
            outMass[1] = i;
        }
    }
    return outMass;
}
public static double[] MinElementAr(double[] mass)
{
    double[] outMass = { mass[0], 0 };
    if (IsUndefinedElement(mass))
    {
        outMass[0] = double.NaN;
        outMass[1] = -1;
        return outMass;
    }
    for (int i = 1; i < mass.Length; i++)
    {
        if (outMass[0] > mass[i])
        {
            outMass[0] = mass[i];
            outMass[1] = i;
        }
    }
    return outMass;
}
}

public static bool IsUndefinedElement(double[] dataAr)
{
    foreach (double el in dataAr)
    {
        if (Double.IsNaN(el)) return true;
    }
    return false;
}

public static bool IsNegativeNumber(double[] dataAr)
{

```

```

        foreach (double el in dataAr)
        {
            if (el <= 0) return true;
        }
        return false;
    }

    public static bool LogRow(ref double[] dataAr, double logBase)
    {
        if (IsNegativeNumber(dataAr) || IsUndefinedElement(dataAr)) return false;
        for (int i = 0; i < dataAr.Length; i++)
        {
            dataAr[i] = Math.Log(dataAr[i], logBase);
        }
        return true;
    }

    public static bool LnRow(ref double[] dataAr)
    {
        if (IsNegativeNumber(dataAr) || IsUndefinedElement(dataAr)) return false;
        for (int i = 0; i < dataAr.Length; i++)
        {
            dataAr[i] = Math.Log(dataAr[i]);
        }
        return true;
    }

    public static bool DifferRow(ref double[] dataAr, int diffRank)
    {
        if (diffRank < 1 || diffRank > dataAr.Length - 1) return false;
        for (int i = dataAr.Length - 1; i > -1; i--)
        {
            if (i < diffRank) dataAr[i] = double.NaN;
            else dataAr[i] = dataAr[i] - dataAr[i - diffRank];
        }
        return true;
    }
}

using System;

namespace ElizabethDataModeling.Algorithms
{
    public class LinearRegression
    {
        private string _paramsAR;
        private string _paramsMA;
        private bool _isAVG;
        private double[] dataAr;
        private Matrix MatrxH;
        private Matrix MatrxResid;
        private Matrix MatrxB;
        private string rowName;

        public LinearRegression() => Clear();

        public void Clear()
        {

```



```

        _paramsAR = "";
        _paramsMA = "";
        _isAVG = false;
        dataAr = null;
        MatrxResid = null;
        MatrxB = null;
        MatrxH = null;
        rowName = null;
    }

    public string ParamsAR
    {
        get => _paramsAR;

        set => _paramsAR = value;
    }

    public string ParamsMA
    {
        get => _paramsMA;

        set => _paramsMA = value;
    }

    public bool IsAVG
    {
        get => _isAVG;

        set => _isAVG = value;
    }

    public int P
    {
        get
        {
            int p = 0;
            if (!String.IsNullOrEmpty(_paramsAR))
            {
                string[] tmp = _paramsAR.Split(';');
                p = int.Parse(tmp[tmp.Length - 1]);
            }
            return p;
        }
    }

    public int Q
    {
        get
        {
            int q = 0;
            if (!String.IsNullOrEmpty(_paramsMA))
            {
                string[] tmp = _paramsMA.Split(';');
                q = int.Parse(tmp[tmp.Length - 1]);
            }
            return q;
        }
    }
}

```

```

    public string RowName
    {
        set => rowName = value;

        get => rowName;
    }

    public double[] DataAr
    {
        set
        {
            dataAr = new double[value.Length];
            value.CopyTo(dataAr, 0);
        }
        get => dataAr;
    }

    public Matrix H
    {
        set => MatrxH = new Matrix(value.Array());

        get => MatrxH;
    }

    public Matrix Resid
    {
        set => MatrxResid = new Matrix(value.Array());

        get => MatrxResid;
    }

    public Matrix B
    {
        set => MatrxB = new Matrix(value.Array());

        get => MatrxB;
    }
}

}
using System.Windows.Forms;

namespace ElizabethDataModeling.Algorithms
{
    public class Matrix
    {
        private double[,] matrix;
        public Matrix(int countRows, int countColumns) => matrix = new double[countRows,
countColumns];

        public Matrix(double[] matrix)
        {
            this.matrix = new double[matrix.Length, 1];
            for (int i = 0; i < matrix.Length; i++)
            {

```

```

        this.matrix[i, 0] = matrix[i];
    }
}
public Matrix(double[,] matrix) => this.matrix = matrix;

public double this[int i, int j]
{
    get => matrix[i, j];

    set => matrix[i, j] = value;
}
static public Matrix Transpose(Matrix matr)
{
    double[,] TAr = new double[matr.CountColumns, matr.CountRows];
    for (ushort i = 0; i < TAr.GetLength(0); i++)
    {
        for (ushort j = 0; j < TAr.GetLength(1); j++)
        {
            TAr[i, j] = matr[j, i];
        }
    }

    Matrix res = new Matrix(TAr);
    return res;
}
public double[,] Array() => matrix;

public double[] RowArray(int indexRow)
{
    double[] rowAr = new double[this.CountColumns];
    for (int i = 0; i < this.CountColumns; i++)
    {
        rowAr[i] = matrix[indexRow, i];
    }
    return rowAr;
}

public int CountRows
{
    get => matrix.GetLength(0);
}
public int CountColumns
{
    get => matrix.GetLength(1);
}

public Matrix Clone(int Si, int Fi, int Sj, int Fj)
{
    Matrix Res = new Matrix(Fi - Si + 1, Fj - Sj + 1);
    for (int i = Si; i <= Fi; i++)
    {
        for (int j = Sj; j <= Fj; j++)
        {
            Res[i - Si, j - Sj] = this[i, j];
        }
    }
    return Res;
}

```

```

}
static public Matrix Inverse(Matrix matr)
{
    if (matr.CountRows != matr.CountColumns)
    {
        MessageBox.Show("Matrix is not squared!");
        return null;
    }

    Matrix res = new Matrix(Method_Haleckogo(matr.Array()));
    return res;
}
static private double[,] Method_Haleckogo(double[,] matrixA)
{
    int n = matrixA.GetLength(0);
    double[,] matrixC = new double[n, n];
    double[,] matrixB = new double[n, n];

    for (int j = 0; j < n; j++)
    {
        for (int i = 0; i < n; i++)
        {
            if (i < j)
            {
                double temp = 0;
                for (int l = 0; l < i; l++)
                {
                    temp += matrixC[i, l] * matrixB[l, j];
                }

                matrixB[i, j] = (matrixA[i, j] - temp) / matrixC[i, i];
            }
        }

        for (int i = 0; i < n; i++)
        {
            if (i >= j)
            {
                double temp = 0;
                for (int l = 0; l < i; l++)
                {
                    temp += matrixC[i, l] * matrixB[l, j];
                }

                matrixC[i, j] = matrixA[i, j] - temp;
            }
        }
    }

    for (int i = 0; i < n; i++)
    {
        matrixB[i, i] = 1;
    }

    double[,] inverseMatrix = new double[n, n];
    int flag = n - 1;

    while (flag != -1)

```

```

{
    for (int l = flag; l > -1; l--)
    {
        if (flag >= l)
        {
            if (l == flag)
            {
                double temp = 0;
                for (int i = l + 1; i < n; i++)
                {
                    temp += matrixC[i, l] * inverseMatrix[flag, i];
                }

                inverseMatrix[flag, l] = (1 - temp) / matrixC[l, l];
            }
            else
            {
                double temp = 0;
                for (int i = l + 1; i < n; i++)
                {
                    temp -= matrixC[i, l] * inverseMatrix[flag, i];
                }

                inverseMatrix[flag, l] = temp / matrixC[l, l];
            }
        }
    }

    for (int k = flag; k > -1; k--)
    {
        if (k < flag)
        {
            double temp = 0;
            for (int i = k + 1; i < n; i++)
            {
                temp -= matrixB[k, i] * inverseMatrix[i, flag];
            }

            inverseMatrix[k, flag] = temp;
        }
    }

    flag--;
}

return inverseMatrix;
}
static public Matrix Increase(Matrix A, Matrix B)
{
    if (A.CountColumns != B.CountRows)
    {
        MessageBox.Show("Different matrix dimensions!");
        return null;
    }
    Matrix res = new Matrix((ushort)A.CountRows, (ushort)B.CountColumns);
    for (ushort i = 0; i < res.CountRows; i++)
    {
        for (ushort j = 0; j < res.CountColumns; j++)

```

```

        {
            for (ushort k = 0; k < A.CountColumns; k++)
            {
                res[i, j] = res[i, j] + A[i, k] * B[k, j];
            }
        }
    }
    return res;
}
static public Matrix Diff(Matrix A, Matrix B)
{
    if (A.CountColumns != B.CountColumns || A.CountRows != B.CountRows)
    {
        MessageBox.Show("Different matrix dimensions!");
        return null;
    }

    Matrix res = new Matrix((ushort)A.CountRows, (ushort)A.CountColumns);
    for (ushort i = 0; i < res.CountRows; i++)
    {
        for (ushort j = 0; j < res.CountColumns; j++)
        {
            res[i, j] = A[i, j] - B[i, j];
        }
    }
    return res;
}
static public Matrix Add(Matrix A, Matrix B)
{
    if (A.CountColumns != B.CountColumns || A.CountRows != B.CountRows)
    {
        MessageBox.Show("Different matrix dimensions!");
        return null;
    }

    for (int i = 0; i < A.CountRows; i++)
    {
        for (int j = 0; j < A.CountColumns; j++)
        {
            A[i, j] = A[i, j] + B[i, j];
        }
    }
    return A;
}
static public Matrix DiagonalMatrix(int size, double diagEl)
{
    Matrix Res = new Matrix(size, size);
    for (int i = 0; i < Res.CountColumns; i++)
    {
        Res[i, i] = diagEl;
    }
    return Res;
}
static public Matrix IncrConst(double c, Matrix A)
{
    for (int i = 0; i < A.CountRows; i++)
    {
        for (int j = 0; j < A.CountColumns; j++)

```

```
        {  
            A[i, j] = c * A[i, j];  
        }  
    }  
    return A;  
}  
  
}
```